

# ABC

N° 42

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

## INFORMATIQUE



Légers, les portables

La banque de l'A.F.P.

Le malin Goupil

Nucléaire en famille

EDITIONS  
**ATLAS**

Grande première.  
Les éditions Atlas présentent :

# LE CINEMA

## GRAND DICTIONNAIRE ILLUSTRE!

Tous les grands noms du Cinéma sont rassemblés pour vous, en ordre alphabétique, et abondamment illustrés de portraits, photos de tournage, scènes inoubliables...

Scénaristes, comédiens, superstars, musiciens, décorateurs, maquilleurs, metteurs en scène, producteurs, composent semaine après semaine le plus grand générique de toute l'histoire du Cinéma.

Une collection indispensable aux amateurs, cinéphiles, et amoureux du 7<sup>e</sup> Art.

**LE CINÉMA, Grand Dictionnaire Illustré, 4000 noms, 2500 photos, 3 volumes. Une production Atlas.**



EDITIONS  
ATLAS



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : (37) 28-10-10. Services administratifs et commerciaux : 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

Publicité : Anne Cayla. Tél. : 202-09-80.

### VENTE AU NUMÉRO

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser aux services commerciaux des ÉDITIONS ATLAS, Z.I. de Lucé, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

### SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

### RELIEZ VOS FASCICULES

Des reliures mobiles permettant de relier 12 fascicules sont en vente chez votre marchand de journaux.

**ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.**

En vente tous les vendredis. Volume IV, n° 42.

ABC INFORMATIQUE est réalisé avec la collaboration de Tristan Mordrel (secrétariat de rédaction), Jean-Pierre Bourcier (coordination), Patrick Bazin, Jean-Paul Moulton, Claire Rémy (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Claire Bischoff (correction).  
Crédit photographique, couverture : Sic-PTT.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : novembre 1984. 28411. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.  
© Editions Atlas, Paris, 1984.

### A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



# Portables légers

**Le premier micro-ordinateur portable lancé en 1981 fut accueilli avec enthousiasme. Depuis, il paraît bien lourd, comparé à la dernière génération d'ordinateurs portables.**

Le sens du terme « portable » a évolué depuis l'arrivée de la dernière génération d'ordinateurs de ce type. En effet, les ordinateurs que l'on qualifiait de portables, il y a quelques années, le sont vraiment devenus. Pour mériter le qualificatif de portables, les machines doivent aujourd'hui intégrer leurs propres unités d'alimentation, d'affichage et de stockage dans un boîtier aux dimensions inférieures à celles d'un annuaire téléphonique.

L'Epson HX-20 fut le premier à offrir ces caractéristiques. Mais son affichage réduit de quatre lignes de 20 caractères lui donne un coup de vieux. Les derniers portables comme le Tandy 100, le NEC PC-8201A ou l'Olivetti M10 offrent à un prix comparable un affichage comptant quatre fois plus de caractères.

Que peuvent donc faire ces ordinateurs? Quels sont les avantages et les inconvénients par rapport aux micros conventionnels? La raison la plus évidente d'acheter un portable est d'avoir accès à la puissance de l'informatique en tout lieu et en tout temps. De nombreuses personnes sont la plupart du temps éloignées de leur ordinateur conventionnel et perdent de précieuses heures dans d'autres bureaux, dans des chambres d'hôtel, dans des aéroports ou dans des trains. L'ordinateur portable permet d'utiliser efficacement tous ces moments jusqu'ici perdus.

La dernière génération de portables permet d'effectuer des travaux scientifiques, techniques, de la comptabilité, de la gestion financière et du traitement de texte; en fait, presque toutes les applications pour lesquelles les ordinateurs conventionnels étaient utilisés.

Les ordinateurs portables offrent généralement au moins trois programmes intégrés, dont un interpréteur BASIC, un programme de traitement de texte et un programme de communications. Le Tandy 100 et l'Olivetti sont aussi munis de programmes intégrés d'adresses et d'agendas qui permettent à l'utilisateur de trouver les adresses, les numéros de téléphone et les rendez-vous du jour.

Le programme de communications est très important puisqu'il permet de communiquer avec d'autres micros et avec des bases de données grâce au réseau téléphonique. Ce programme peut également transformer le portable en un terminal de télex ou en un récepteur ou émetteur de courrier électronique. Évidemment, il est nécessaire d'utiliser pour cela un modem ou un coupleur acoustique.



## En déplacement

L'utilisation de l'ordinateur au cours de voyages devient de plus en plus courante, principalement chez les hommes d'affaires. Certains utilisent leur portable pour effectuer quelques minutes de traitement de texte pendant leurs trajets en train ou en avion. D'autres, comme les représentants, inventent de nouvelles méthodes de travail en apportant un ordinateur chez leurs clients, afin de produire sur place des devis qui, normalement, auraient dû nécessiter des jours de préparation. Les hommes d'affaires en déplacement peuvent transmettre des données à leur siège social à l'aide d'un modem et du réseau téléphonique et, à la fin de la journée, ils peuvent regagner leur bureau et envoyer les données directement vers un plus gros ordinateur. (Cl. Tony Sleep.)

Les ordinateurs portables les plus coûteux, comme le Sharp-5000 et l'Epson PX-8, utilisent les systèmes d'exploitation MS-DOS et CP/M. Ils sont donc en mesure d'exécuter une vaste gamme de programmes de gestion.

L'Epson PX-8 est livré avec le populaire programme de traitement de texte Wordstar déjà installé en ROM. Le Sharp utilise des cartouches de mémoire à bulles qui donnent 128 K de mémoire additionnelle. Ces cartouches gèrent les données aussi rapidement que des lecteurs de disquette.

## Informatique en vol

Officiellement, l'Association d'Aviation civile est d'avis que les équipements électroniques alimentés par piles peuvent causer une interférence sur les commandes de pilotage des avions. Mais les compagnies aériennes interprètent très différemment ces recommandations.



Les compagnies Lufthansa et Qantas interdisent catégoriquement tout équipement électronique à bord. Pan Am, l'une des plus grosses compagnies aériennes américaines, n'autorise pas les radios et les magnétophones mais permet les jeux électroniques et les ordinateurs.



Les portables moins coûteux utilisent des programmes d'application dédiés qui sont généralement chargés dans la RAM à partir d'une unité à cassette. Ce processus est beaucoup plus lent qu'un chargement à partir d'une cartouche de mémoire à bulles ou à partir d'un lecteur de disquette. Le NEC PC-8201A est livré avec une cassette renfermant plusieurs programmes d'application. Citons parmi ceux-là une calculatrice à mémoire, un formateur de texte, un gestionnaire de portefeuille boursier et un évaluateur de prêts. La calculatrice à mémoire transforme la machine en une calculatrice qui peut mémoriser jusqu'à 99 entrées. Le formateur de texte prépare l'impression de fichiers qui ont été entrés par le programme de traitement de texte en spécifiant des largeurs de marge, en divisant le texte en pages, en attribuant des numéros de pages, etc. Le gestionnaire de portefeuille boursier s'adresse aux personnes qui désirent évaluer le rendement de leurs actions. Ce programme analyse un portefeuille comptant jusqu'à 50 investissements.

Comme pour tout autre ordinateur, le micro portable peut être connecté à des périphériques (une imprimante, une unité à cassette et un modem). En plus des facteurs dimensions et poids, le véritable portable se caractérise par

son autonomie d'alimentation, par la présence d'un affichage intégré, et par la présence en ROM d'un programme de traitement de texte et d'un programme de communications.

Des machines comme l'Apple IIc ou comme l'Apricot sont annoncées comme portables. Mais elles ne peuvent être utilisées dans des moyens de transport puisqu'elles doivent être branchées au secteur, connectées à un moniteur, et qu'elles doivent charger leurs programmes en RAM à partir d'une disquette. A l'exception de leur petite taille et de leur légèreté, elles ont beaucoup plus de points communs avec le micro de table conventionnel qu'avec le véritable ordinateur portable fonctionnant à piles.

En plus de leurs piles principales, les ordinateurs portables sont munis de petites piles de nickel-cadmium qui assurent une alimentation d'urgence. Ce dispositif est essentiel puisque toutes les données seraient perdues en cas d'épuisement des piles.

La plupart des portables possèdent une interface de lecteur de code à barres afin de pouvoir être utilisés pour la gestion de stocks. Parmi les machines représentées dans ces pages, le Tandy modèle 100, le NEC PC-8201A et l'Olivetti M10 sont tous munis d'un lecteur de code à barres, mais le Casio FP-200 ne l'est pas. Les trois premières machines sont en fait très similaires à de nombreux égards, puisqu'elles sont toutes fabriquées dans la même usine japonaise. La différence significative entre elles réside dans le fait que l'Olivetti a un écran inclinable, que le logiciel et les mémoires intégrés du NEC sont



**Epson HX-20**  
Malgré la petite dimension de son écran, le HX-20 offre l'avantage d'une unité à cassette et d'une petite imprimante intégrées. Un modeste programme de traitement de texte est aussi inclus.



**Casio FP-200**  
Le Casio est l'ordinateur portable le moins cher, mais il ne comporte pas de programme de traitement de texte intégré. Il offre, à la place, une sorte de tableur.





moins importants et que le Tandy et l'Olivetti ne peuvent aller au-delà de 32 K tandis que le NEC peut être porté à 64 K. Le NEC peut aussi utiliser des cartouches de mémoire de 32 K interchangeables qui conservent leurs données même lorsqu'elles sont retirées de l'ordinateur. Toutes les extensions de mémoire pour ordinateurs portables sont coûteuses en raison du type de puce utilisé.

Contrairement aux plus gros micros munis de claviers de type machine à écrire, les ordinateurs

portables ne possèdent que des claviers plats qui sont plus fatigants à utiliser. Les portables de bas de gamme ne peuvent exécuter les populaires programmes de gestion livrés sur disquettes.

Cela dit, l'ordinateur portable a vraiment sa place sur le marché. L'utilisation largement répandue des micros fait de plus en plus découvrir aux gens comment l'informatique peut les aider dans leur vie quotidienne. L'ordinateur portable permet de disposer de cet outil en tout lieu. Bientôt, les ordinateurs portables seront tout aussi répandus que les calculatrices de poche.

#### Epson PX-8

Cette machine peut utiliser un logiciel de gestion de type CP/M, dont le fameux Wordstar qui est livré avec l'ordinateur.

#### Tandy TRS-80

modèle 100/NEC  
PC-8201A/Olivetti M10

Ces machines sont des versions différentes du même micro. Elles partagent un excellent programme de traitement de texte et une vaste gamme d'interfaces. Un modem Olivetti alimenté par piles est aussi présenté.

Modèle	Prix	Mémoire standard	Mémoire maximum	Taille écran	Poids
Casio FP-200	****	6 K	32 K	8 x 20	1,400 kg
Epson HX-20	*****	16 K	32 K	4 x 20	1,800 kg
Epson PX-8	*****	64 K	64 K	8 x 80	2,300 kg
NEC PC-8201	*****	16 K	64 + 32 K	8 x 40	1,800 kg
Olivetti M10	*****	8 K	32 K	8 x 40	1,800 kg
Tandy TRS-80 modèle 100	*****	8 K	32 K	8 x 40	1,800 kg



# Premières impressions

**Les capacités graphiques des imprimantes à matrice de points sont souvent négligées. Nous montrons dans cet article comment initialiser une imprimante pour produire un graphisme attrayant.**

La plupart des micro-ordinateurs disposent d'un graphisme haute résolution qui permet de construire des images à partir de caractères graphiques. Les caractères graphiques ont la même taille que les caractères conventionnels de type texte. Leur code est supérieur à 127 puisque les numéros 0 à 127 sont réservés au jeu de caractères ASCII. `PRINT CHR$(90)` affichera un caractère ASCII à l'écran, « Z » en l'occurrence, et `PRINT CHR$(128)` affichera un caractère graphique — un rectangle noir pour un micro Dragon.

Pour imprimer la lettre « Z », il faut taper `LPRINT CHR$(90)`, aussi pourrait-on penser que pour imprimer un rectangle noir, il conviendrait de manière similaire de lancer la commande `LPRINT CHR$(128)`. Cela n'est malheureusement pas le cas, car le caractère codé 127 varie énormément d'un micro à l'autre. Les fabricants d'imprimantes ne peuvent pas, de toute évidence, prévoir une machine par type de micro-ordinateur. Alors, ils copient le jeu de caractères ASCII sur les caractères 128-255, ou programment leurs propres caractères graphiques.

Les imprimantes Epson ne comportent pas de caractères graphiques. Aussi devrez-vous modifier les codes ASCII standard pour créer vos

propres caractères graphiques. Vous y parviendrez en envoyant les codes d'échappement appropriés à l'imprimante.

Le graphisme haute résolution s'obtient à partir de petits points, ou pixels, et non de caractères entiers. De manière similaire, l'impression haute résolution utilise l'impression de petits points. La tête d'impression d'une imprimante à matrice de points comporte un certain nombre d'aiguilles disposées verticalement qui se déplacent sur le papier. Les caractères sont généralement constitués d'une grille de points (par exemple, de 8 sur 8). Il est néanmoins possible de produire des motifs ou caractères graphiques particuliers en contrôlant individuellement les aiguilles.

Il faut d'abord mettre l'imprimante en mode graphique. Comme pour tout autre type d'impression, on le fait en envoyant un code d'échappement approprié au type de l'imprimante utilisée. Sur l'Epson FX-80, par exemple, les instructions nécessaires sont :

```
LPRINT CHR$(27); «K»;CHR$(N1);(N2);
```

La lettre «K» spécifie le mode graphique et les nombres (N1) et (N2) donnent la largeur de chaque ligne de graphisme (le nombre de points sur la page).

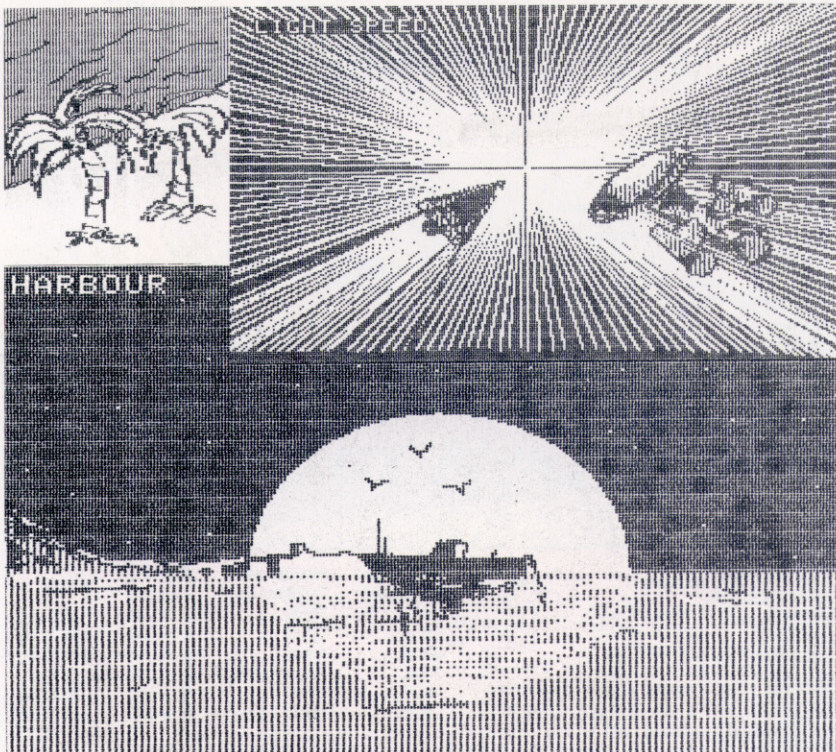
En mode graphique standard, l'imprimante FX-80 peut imprimer un maximum de 480 points par ligne. D'autres modes permettent des résolutions comprises entre 576 et 1920 points. Si nous voulons pouvoir utiliser toute la ligne, 480 sera la bonne longueur de ligne. Deux nombres sont nécessaires pour l'indiquer, parce que la valeur maximale pour un nombre est de 255. Aussi le deuxième nombre (N2) sera multiplié par 256 et ajouté au premier (N1). Pour 480, les nombres sont 1 et 224 ( $480 = 256 \times 1 + 224$ ). L'instruction pour l'imprimante Epson FX-80 sera donc :

```
LPRINT CHR$(27);«K»;CHR$(224);CHR$(1);
```

Ayant programmé l'imprimante pour la longueur de ligne graphique, il nous faut maintenant transmettre les données graphiques. Bien qu'il y ait neuf aiguilles sur la tête d'impression, seules les huit aiguilles supérieures peuvent être utilisées pour la plupart des modes graphiques. Les aiguilles sont numérotées à partir du bas : 1, 2, 4, 8, 16, 32, 64 et 128. Les données pour les huit aiguilles peuvent donc prendre la forme d'un seul nombre compris entre 0 et 255. Ce dernier est envoyé à l'imprimante par

### Recopie d'écran

Ces motifs ont été conçus à l'écran au moyen d'une table graphique. Le contenu de l'écran a été ensuite recopié sur l'Epson FX-80 pour montrer les capacités graphiques des imprimantes à matrice de points (appelées également « imprimantes à aiguilles »).  
(Cl. Simon Dayton.)



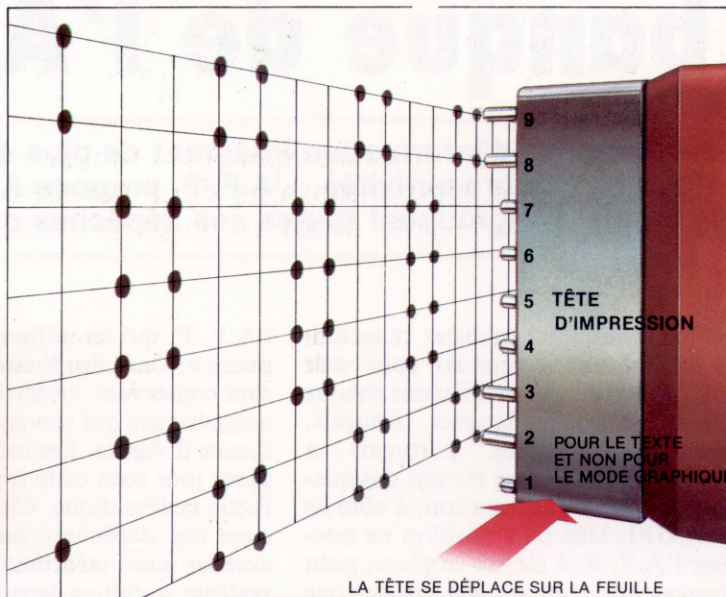


**Tracé d'aiguille**

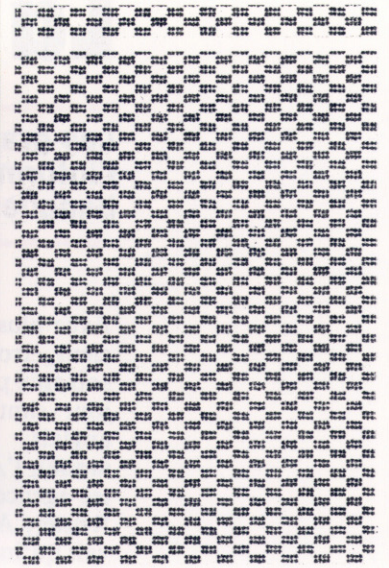
Le motif a été produit sur une imprimante à matrice de points en envoyant à la tête d'impression une suite alternée des nombres 195 et 60. L'avance du papier permet à la ligne suivante de combler les espaces laissés par l'aiguille n° 1.

N° AIGUILLE	VALEUR AIGUILLE		
9	128	●	○
8	64	●	○
7	32	○	●
6	16	○	●
5	8	○	●
4	4	○	●
3	2	●	○
2	1	●	○
		195	60

● L'AIGUILLE IMPRIME  
○ L'AIGUILLE N'IMPRIME PAS



LA TÊTE SE DÉPLACE SUR LA FEUILLE



Steve Cross

LPRINT CHR\$(X), avec X pour le nombre. Si nous ne voulons utiliser que l'aiguille inférieure, nous indiquerons CHR\$(1), l'aiguille supérieure, CHR\$(128), etc. Pour impliquer plusieurs aiguilles, nous ajouterons simplement les nombres pour chacune d'elle. Cette démarche est ensuite répétée pour les 480 positions de la ligne.

Dans la figure ci-dessus, il y a deux motifs ou configurations d'aiguilles, CHR\$(195) et CHR\$(60). Pour imprimer les quatre premières positions de la ligne, nous ferons :

```
LPRINT CHR$(195);CHR$(195);CHR$(60);CHR$(60);
```

Après cela, le motif se répète, et une boucle FOR...NEXT suffit pour coder la fin de la ligne.

Il est important de comprendre que CHR\$(60), dans cet exemple, ne demande pas à l'imprimante d'imprimer un caractère de code ASCII 60, mais qu'il s'agit d'un mode de représentation des données par les aiguilles de la tête d'impression. L'imprimante les reconnaît dans la mesure où nous avons transmis au préalable la séquence CHR\$(27);«K» qui active le mode graphique.

Cette méthode d'impression s'appelle aussi impression binaire de l'image. A côté de l'Epson FX-80, les autres imprimantes utilisent une méthode similaire.

En balayant l'écran dans sa totalité, le programme repère les pixels. Lorsqu'une position en contient un, nous devons demander à la tête d'impression d'actionner une aiguille à l'emplacement correspondant sur le papier. Le balayage est assuré par la fonction POINT(x,y) ou par des commandes semblables disponibles sur la plupart des micros. Lorsqu'un pixel est activé ou allumé, la fonction POINT(x,y) prend la valeur 1; dans le cas contraire, elle prend la valeur 0. Les différentes résolutions d'écran pour les micros vous obligeront probablement à faire des adaptations.

Vous vous demandez peut-être comment un programme de recopie d'écran gère l'affichage couleurs. La solution est d'utiliser des motifs de

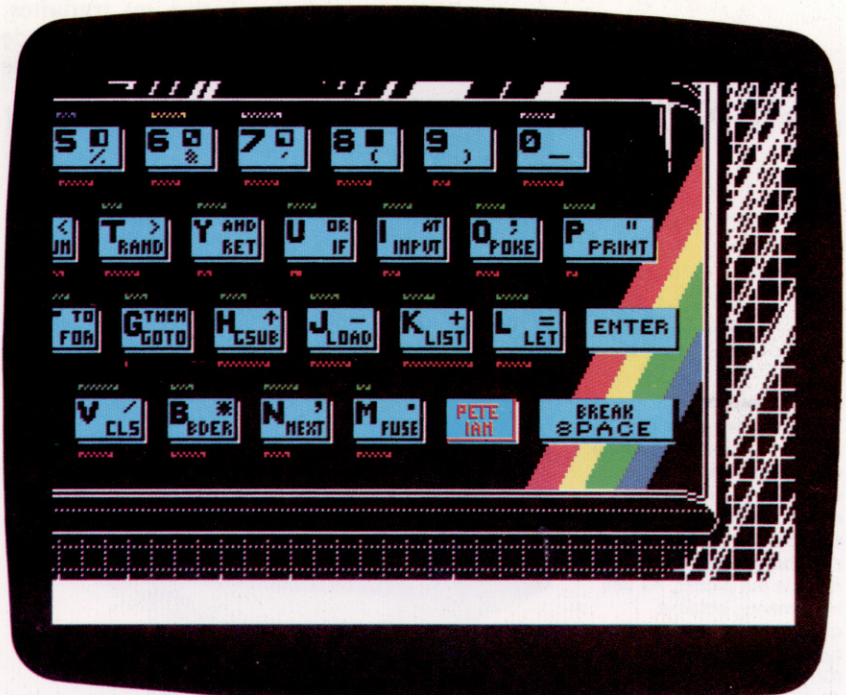
points différents pour chaque couleur. Un pixel noir pourra utiliser quatre points sous forme de carré, un rouge, deux points et un blanc sera représenté par l'absence de points. La fonction POINT(x,y) produit un nombre différent selon la couleur du pixel et est donc toujours utilisable.

Les programmes de recopie d'écran sont généralement ajoutés à la fin des programmes graphiques dans la forme de sous-programmes. Pour recopier l'image à l'écran sur l'imprimante, vous pourrez par exemple avoir à frapper la touche «P», ce qui fera passer l'exécution du programme de création graphique sur le sous-programme d'impression. Un programme de recopie d'écran en BASIC tend à être un peu lent. Les versions de ces programmes utilisant du code machine sont légèrement plus rapides.

**Jets de couleurs**

Cette image du clavier du Spectrum a été obtenue sur une imprimante à jets d'encre de couleurs. Il s'agit en fait d'une imprimante à matrice de points sur laquelle les aiguilles ont été remplacées par des jets d'encre.

(Cl. Ian McKinnell/ Dimension Graphics.)





# La banque de l'A.F.-P.

**Les grandes agences d'information intègrent de plus en plus l'informatique dans leurs activités. L'A.F.-P. propose Agora, une banque de données regroupant toutes ses dépêches de l'année.**

Avec plus de deux mille journalistes et techniciens, trois cents bureaux répartis dans cent soixante pays, 10 000 dépêches diffusées chaque jour simultanément en six langues (français, anglais, allemand, espagnol, portugais et arabe), l'Agence France-Presse est une des quatre agences mondiales d'information, à côté de Reuter, AP et UPI. Une telle position ne pouvait qu'inciter l'A.F.-P. à mettre en place, pour la presse francophone du monde entier, une véritable banque de données internationales en langue française. L'initiative était prise en 1981 et, ainsi, le projet Agora allait pouvoir naître.

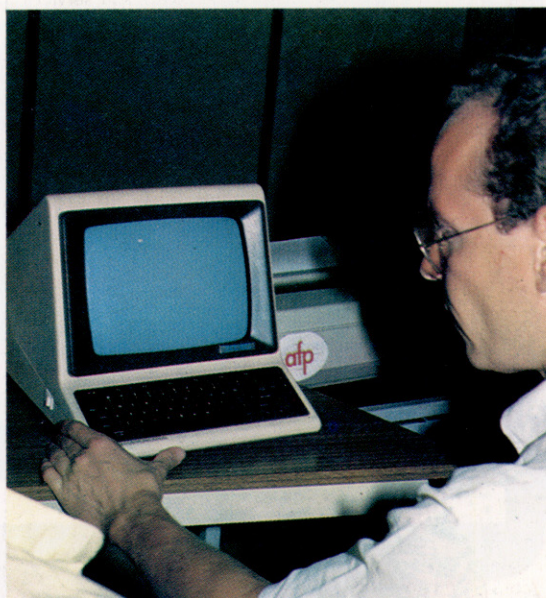
Pour mettre en place cette banque de données, l'A.F.-P. a fait appel à un partenaire technique — le G.C.A.M. —, une des plus importantes firmes françaises spécialisées dans le service informatique. Résultat : Agora permet aux médias, aux entreprises, aux organismes publics et internationaux de retrouver instantanément toute dépêche émise par l'A.F.-P. au cours des douze derniers mois. Son système d'interrogation en langage naturel est particulièrement simple; il a été conçu pour être utilisé par une personne non spécialiste en moins d'une heure.

La collecte mondiale de l'information est centralisée sur un ordinateur au siège de l'agence à Paris. Toutes les informations y sont relues et, éventuellement, mises en forme ou traduites. Elles sont aussitôt diffusées dans le monde entier vers les téléscripteurs des clients de

l'A.F.-P. qui les utilisent comme « matière première ». Dans l'ordinateur central, ces dépêches sont conservées, en texte intégral, sur les bandes magnétiques qui servent à la mise à jour quotidienne d'Agora. Les informations stockées jour après jour sous cette forme sont répertoriées de façon systématique. Chaque mot écrit dans chacune des dépêches constitue un mot clé pour accéder à ces informations. L'interrogation du système se fait en langage libre selon le logiciel américain Stairs.

Agora peut être considérée comme le prolongement naturel des services classiques de l'A.F.-P. Les dépêches d'agence présentent d'ailleurs d'excellentes prédispositions pour l'interrogation sur banques de données. La dépêche d'agence est en effet concise et courte. On parle beaucoup actuellement d'« écriture télématique », tout en s'interrogeant sur la meilleure façon de rédiger une information destinée à être lue, non plus dans les colonnes d'un journal, mais sur le petit écran d'un terminal informatique. Il nous semble précisément que la dépêche d'agence y est assez bien adaptée. Le « lead », c'est-à-dire les quatre premières lignes de la dépêche, comporte systématiquement les éléments importants et nouveaux de l'information rapportée. Dans ces quatre premières lignes, le lecteur trouve un condensé de l'information qui sera développée dans les paragraphes suivants. Il détermine immédiatement si l'information est ou non susceptible de l'intéresser. Les paragraphes suivants développent les éléments de l'information dans l'ordre de leur importance. Cette technique facilite la consultation des dépêches sur écran.

Actuellement, Agora permet à l'utilisateur d'accéder aux informations diffusées par l'A.F.-P. en langue française au cours des douze derniers mois. Ces informations sont réparties dans deux fichiers différents : Agora-Général et Agora-Économie. Agora-Général conserve toute l'actualité mondiale (politique et relations internationales, grands reportages, informations scientifiques et culturelles, faits divers, au total 250 000 dépêches (50 millions de mots ou 450 millions de caractères). Une place à part est faite aux informations sur l'économie, les finances et le monde des affaires. Elles font l'objet de développements spécifiques plus détaillés, qui sont regroupés dans Agora-Économie. Ces dépêches concernent les conférences internationales, les grands contrats, les chiffres clés de l'économie mondiale, les



## Vidéotex Minitel

Le Minitel peut être utilisé pour l'interrogation d'Agora. Ce terminal se prête cependant assez mal aux recherches et, surtout, à la visualisation d'informations développées. En outre, la recopie sur papier, via une imprimante, soulève quelques difficultés sur ce type de matériel.  
(Cl. Renée Panier/  
Arch. Éd. Atlas.)





marchés financiers, etc. Elles représentent au total 90 000 dépêches (18 millions de mots ou 162 millions de caractères).

La mise à jour d'Agora est effectuée quotidiennement. Les utilisateurs retrouvent chaque matin les informations diffusées par l'A.F.-P. jusqu'à la veille au soir. Dans le même temps, le système est régulièrement allégé des informations les plus anciennes, celles qui ont plus d'un an. Si les informations d'actualité vieillissent assez rapidement dans leur ensemble, certaines d'entre elles présentent un intérêt plus durable, voire permanent (biographies, composition de gouvernements, fiches techniques en tous genres en sont des exemples). Une banque de données spécifique est en cours de constitution pour préserver les informations de cette nature : Agora-Documentaire. Cette dernière conserve ses informations sans limitation de durée. Elle est alimentée à partir d'une sélection de dépêches effectuée dans le corps d'Agora-Général et Agora-Économie.

## Le système Agora

L'accès au système Agora se fait par téléphone, *via* le réseau Euronet, à partir de n'importe quel terminal compatible télétype. La vitesse d'impression des dépêches demandées peut être de 300 bauds ou 1 200 bauds selon les appareils. Chaque utilisateur dispose de son propre code d'accès au système. Les recherches sont facturées à la durée de connexion (520 F pour une heure). Le client d'Agora ne supporte aucun coût d'abonnement. L'interrogation du système se fait en langage libre : les recherches peuvent porter sur n'importe quel mot, à l'exception de quelques articles indéfinis et pronoms considérés comme non significatifs. L'utilisateur d'Agora formule ses questions en combinant les mots les plus représentatifs de l'information qu'il recherche. Noms de personnes ou d'organismes, noms de lieux sont particulièrement efficaces. L'ordinateur affiche d'abord le nombre des dépêches qui répondent à la question posée. Si ce nombre paraît très élevé, l'utilisateur peut réduire le champ de sa question en ajoutant de nouveaux critères de sélection. A tout moment, une simple commande permet l'impression du texte intégral des dépêches repérées par le système.

Pour affiner sa recherche, l'utilisateur d'Agora dispose d'un certain nombre d'opérateurs : ET, pour que les mots figurent dans le même texte (pétrole ET Venezuela); OU, pour que l'un ou l'autre des mots apparaisse dans la dépêche (augmentation OU hausse); ADJ, pour repérer les mots adjacents (Premier ADJ ministre); PHR, pour que les mots apparaissent dans la même phrase (terrorisme PHR Italie); enfin, le système de la troncature permet de repérer automatiquement les variations morphologiques d'un même mot, au masculin et au féminin, au singulier et au pluriel, etc. En outre, il est possible de préciser à l'ordinateur l'emplacement des mots recherchés.



L'affichage sur l'écran des seuls « leads » de chaque dépêche permet d'effectuer un premier et rapide survol des informations disponibles. Il reste alors à demander l'impression en texte intégral des informations qui paraissent les plus intéressantes.

**Terminaux standard ASCII**  
Ci-dessus : le « desk » central de l'A.F.-P., les journalistes travaillent en liaison constante avec les banques de données Agora. Agora est accessible à partir de n'importe quel terminal ASCII ou Vidéotex (le Minitel).

A l'achat, il faut prévoir un investissement minimal de 14 000 F pour une configuration complète (terminal vidéo, modem et imprimante). Le terminal Matra TTE 415 est particulièrement bien adapté à l'interrogation de banques de données en texte intégral de type Agora. Une formule de location est proposée par Locatel, en accord avec l'A.F.-P., au tarif de 450 F par mois pour une configuration complète, terminal Matra TTE 415 et imprimante Microline. (Cl. Renée Panier/ Arch. Éd. Atlas.)

### Les quatre banques de données Agora

Agora est subdivisée en quatre banques de données accessibles séparément. Une commande simple, que l'on peut effectuer en cours d'interrogation, permet de passer facilement de l'une à l'autre. Un abonnement à Agora donne accès automatiquement à ces quatre sous-ensembles. Les procédures d'interrogation sont identiques.

#### Agora-Général (AGRA)

L'ensemble des informations, françaises et internationales, diffusées par l'A.F.-P. dans son service général. Informations ponctuelles ou développées sur tous les sujets d'actualité, à l'exception du sport. Politique, diplomatie, économie générale, culture, sciences, faits divers.

#### Agora-Économie (AECO)

Toutes les informations économiques et financières, françaises et internationales. Pays industriels et pays en développement; industrie, commerce, agriculture, transports, équipement, énergie, banque, bourse, sociétés, échanges internationaux; informations sociales.

#### Agora-Sports (ASPO)

Les informations sportives de l'A.P.-F. en provenance du monde entier. Couverture de toutes les disciplines sportives, de toutes les grandes rencontres nationales et internationales. Reportages, résultats, synthèses, chronologies, portraits.

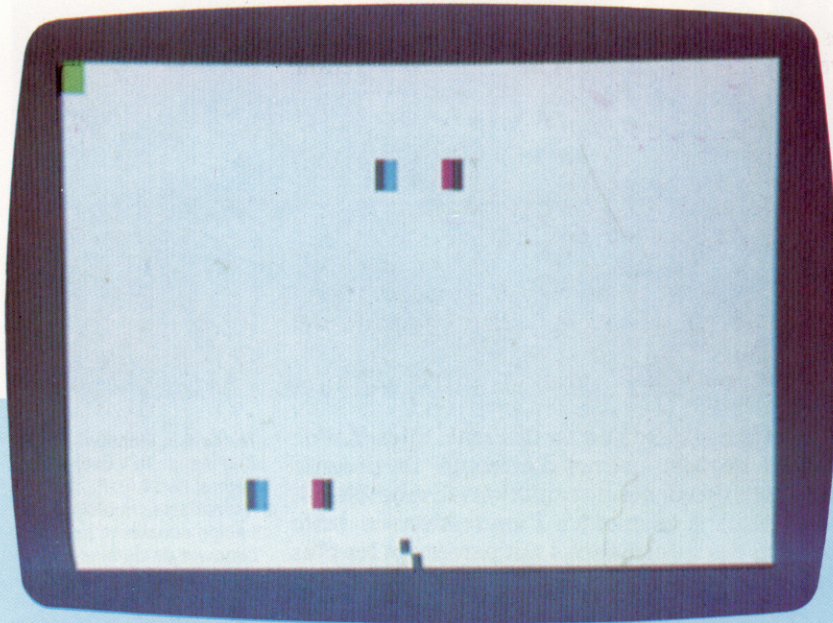
#### Agora-Documentaire (ADOC)

Une sélection des informations les plus « durables » émises par l'A.F.-P., conservées sans limitation de temps. Biographies, chronologies de grands événements, synthèses, bilans, listes de gouvernements, fiches techniques, « encadrés », etc.



# Slalom

Les jeux de sport amusent petits et grands. Le goût de la performance reste vivace, même devant un écran. Pierre Monsaud a écrit ce jeu pour le micro-ordinateur Dragon.



Partez aux sports d'hiver sans risque de vous casser une jambe! Lancez-vous du haut de la piste et essayez de passer le plus grand nombre possible de portes sans heurter les piquets. Frappez n'importe quelle touche pour changer de direction.

```
10 REM *****
20 REM * SLALOM *
30 REM *****
34 REM
35 REM INITIALISATION
36 REM
38 REM TABLEAU DES POSITIONS
39 REM DU SKIEUR
40 DIM S$(1)
50 FOR I=1 TO 32
60 E#=E#+CHR$(207)
70 NEXT I
79 REM SKIEUR ALLANT A GAUCHE
80 S$(0)=CHR$(201)
89 REM SKIEUR ALLANT A DROITE
90 S$(1)=CHR$(198)
99 REM FOND BLANC
100 CLS 5
108 REM DIRECTION INITIALE :
109 REM          GAUCHE
110 D=-1
118 REM POSITION INITIALE
119 REM DU SKIEUR
120 J=16
129 REM DESSIN DES PORTES
130 P#=CHR$(181)+CHR$(207)+CHR$(
207)+CHR$(170)
134 REM
135 REM BOUCLE PRINCIPALE
136 REM
140 FOR K=1 TO 300
145 REM CALCUL DES COORDONNEES
146 REM DU SKIEUR
150 Y=INT(J/32)*2
160 X=(J-16*Y)*2
165 REM SKIEUR AU NIVEAU D'UNE
166 REM PORTE?
170 IF K>=16 AND (K-5)/10=INT((K
-5)/10) THEN GOSUB 330
179 REM AFFICHAGE D'UNE PORTE?
180 IF K<284 AND K/10=INT(K/10)
THEN GOSUB 350
189 REM MOUVEMENT DU SKIEUR
190 IF INKEY$("<>") THEN D=-D
200 J=J+D
210 IF J<2 THEN J=2
220 IF J>29 THEN J=29
230 PRINT@ 511,E#;
240 PRINT@ J,S$(D/2+0.5);
250 NEXT K
254 REM
255 REM FIN DU PARCOURS
256 REM
260 PRINT@ 164,"PORTE(S) RATEE(S
) :";T;
270 PRINT@ 229,"UNE AUTRE DESCEN
TE ?";
280 D$=INKEY$
290 IF D$="" THEN 280
300 IF D$("<>N") THEN RUN
310 CLS
320 END
324 REM
325 REM PORTE RATEE?
326 REM
330 IF POINT(X-2,Y)<>0 OR POINT(
X+4,Y)<>3 THEN IF POINT(X-4,Y)<>
0 OR POINT(X+2,Y)<>3 THEN T=T+1:
SOUND 1,1
340 RETURN
344 REM
345 REM AFFICHAGE D'UNE PORTE
346 REM
350 P1=RND(3)-2
360 P=P-6*P1
370 IF P<482 THEN P=488
380 IF P>506 THEN P=500
390 PRINT@ P,P#;
400 RETURN
```



# Goupil le malin

**Très utilisés, entre autres, dans les clubs Microtel, les micro-ordinateurs Goupil fabriqués par S.M.T. ont acquis une grande notoriété. Leur modularité est un atout essentiel.**

En 1979, la Société de Micro-informatique et Télécommunications — S.M.T. — prenait naissance en France avec pour objectif de concevoir et de diffuser des micro-ordinateurs destinés à des usages professionnels ou grand public. La puissante Délégation Générale des Télécommunications lui apportait un soutien efficace pour qu'elle introduise sur le marché toute une gamme d'ordinateurs dénommés Goupil. De nombreuses fées entouraient aussi de leurs soins les développements de ces matériels « à la tête de renard » puisque l'on trouve comme partenaires de S.M.T., des entreprises comme ELF Aquitaine, Control Data, Locatel ou Nathan relayées sur le terrain par les clubs Microtel.

L'idée qui domine lors de la conception de la gamme des Goupil est la modularité. La même machine qui permet au professeur de parfaire son enseignement au moyen de didacticiels doit aussi permettre plus tard à l'élève de se détendre en s'initiant à la création graphique ou musicale, ou en jouant aux jeux les plus divers. Aussi existe-t-il dix configurations standard de Goupil, cinq configurations de base et cinq configurations étendues. Quelle que soit la configuration choisie, il est possible de l'étendre en acquérant tel ou tel périphérique, ou telle ou telle carte électronique.

Les configurations de base sont toutes équipées d'une mémoire RAM de 64 K et d'une carte bus « fond de panier » à 7 connecteurs.

Les configurations étendues sont équipées d'une mémoire RAM de 64 K, 128 K ou 256 K extensible à près d'1 M octet, d'une carte bus « fond de panier » à 7 ou 12 connecteurs et d'une carte contrôleur disques DMA (sauf pour la configuration 8 avec lecteurs 5", celle-ci ne nécessitant qu'une alimentation 50 W). Il faut noter que si, pour la plupart de ses applications, Goupil fonctionne en monoposte, dans sa configuration 9, il offre également toutes les possibilités d'un ordinateur multitâche extensible multiposte (sous-système d'exploitation Uniflex © TSC, dérivé de Unix © Bell Laboratories).

Modulaire, Goupil l'est doublement. D'une part, à partir de ses configurations de base, l'utilisateur peut acquérir progressivement différents périphériques qui étendent ses possibilités; il faut remarquer que son clavier, son écran vidéo et son boîtier-lecteur de disquette peuvent s'attacher à la console ou en rester détachés, ce qui permet de les disposer comme on l'entend. D'autre part, et c'est là l'un des points

essentiels, l'utilisateur peut accroître progressivement les capacités de la machine en acquérant de nouvelles cartes électroniques qui viennent s'enficher à l'arrière de la console.

Qu'il s'agisse de périphériques ou de cartes électroniques, la modularité de Goupil permet à l'utilisateur d'échelonner ses investissements et de disposer d'un outil de travail répondant à ses besoins.

Le clavier se compose de 101 touches réparties en trois blocs séparés auxquels s'ajoute une rangée supérieure :

- le premier bloc forme un clavier AZERTY (ou QWERTY) identique à celui d'une machine à écrire; il comporte les minuscules accentuées et tous les signes de ponctuation;

- le deuxième bloc regroupe les touches servant à déplacer le curseur sur l'écran vidéo (applications en traitement de texte et en édition);

- le troisième bloc est un clavier numérique (applications en comptabilité et en gestion);

- la rangée supérieure présente les touches de fonction, utiles pour certains logiciels spécifiques.

L'écran vidéo permet d'afficher 25 lignes de 80 caractères. Il est équipé d'un filtre anti-reflet et possède les fonctions d'affichage suivantes : inversion, clignotement, demi-brillance, soulignement, caractères gras, zone masquée (où l'on peut accumuler des caractères sans les afficher sur l'écran). A l'initialisation, il offre 128 caractères standard prédéfinis et l'utilisa-

## Systèmes d'exploitation à la demande

La liste ci-dessous énumère les différents systèmes d'exploitation que Goupil peut mettre en œuvre, et les différents langages de programmation qu'ils rendent accessibles (liste non exhaustive) :

Sous FLEX-9 (© TSC) : Macro Assembleur 6809; SBasic, Logo, Pascal, Fortran, Forth, Lisp, LSE.

Sous UNIFLEX (© TSC) : Macro Assembleur 6809; Basic, Fortran, Pascal, C, Cobol.

Sous UCSD (© Softech) : Macro Assembleur 6809; Pascal, Fortran, Basic.

Sous CP/M ou CP/M 86 (© Digital research) : Macro Assembleur; MBasic 80, CBasic 80, Fortran 80, APL, Pascal MT+, CIS Cobol, LSE.

Sous MS/DOS (© Microsoft) : Assembleur Masm 8088; SBasic, MSBasic, MSBascom, MSFortran, MSCobol, MSPascal, C, Cobol, MBP.

Cet ensemble ouvre à l'utilisateur l'accès d'une très vaste bibliothèque de logiciels d'application, capables de répondre aux besoins les plus divers.



teur peut créer, au moyen de plusieurs logiciels utilitaires, les polices de caractères qu'il désire.

A noter que les affichages de textes et les affichages semi-graphiques de Goupil répondent à la norme Vidéotex européenne.

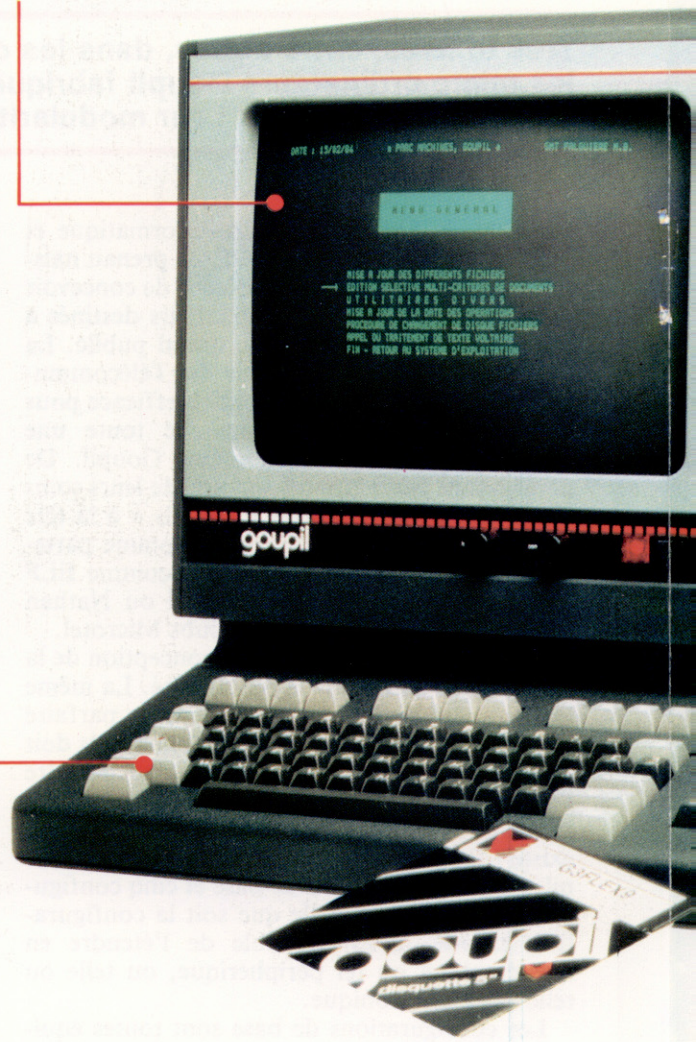
Autre particularité du Goupil : il accepte toute la gamme des mémoires de masse disponible, depuis les disquettes 5 pouces double face double densité jusqu'aux disques durs de 2 fois 10 M octets.

Cette ouverture des machines Goupil est essentiellement due à la modularité de l'électronique utilisée. L'unité centrale peut être bâtie autour des processeurs 6809 Motorola, Z80 Zilog ou 8088 Intel (il faut noter d'ailleurs que deux cartes électroniques correspondant à deux de ces microprocesseurs peuvent être placées ensemble dans la machine), elle offre à l'utilisateur la possibilité de choisir entre plusieurs systèmes d'exploitation et met à sa disposition un nombre considérable de logiciels de base et d'application.

Tous ces atouts ont pourtant une face cachée : si les Goupil s'adaptent aux besoins des professionnels comme à ceux des familles, ils restent des produits chers, donc difficilement accessibles pour le plus grand nombre.

**Vidéo Goupil 25 x 80**

Boîtier écran 25 x 80 professionnel  
12 pouces avec filtre et carte  
contrôleur vidéo 25 x 80.



**Le clavier**

Il se compose de 101 touches réparties en trois blocs séparés auxquels s'ajoute une rangée supérieure.

**Carte mémoire RAM**

Chaque « puce » dorée peut contenir plus de 8 000 caractères.

**Carte microprocesseur**

« Cœur » de la machine, le microprocesseur (ici un 6809) gère l'ensemble du micro-ordinateur.

**Carte d'affichage**

Elle permet d'afficher sur un écran monochrome ou couleurs soit du texte (25 lignes de 80 caractères), soit du graphique (512 par 256 points).

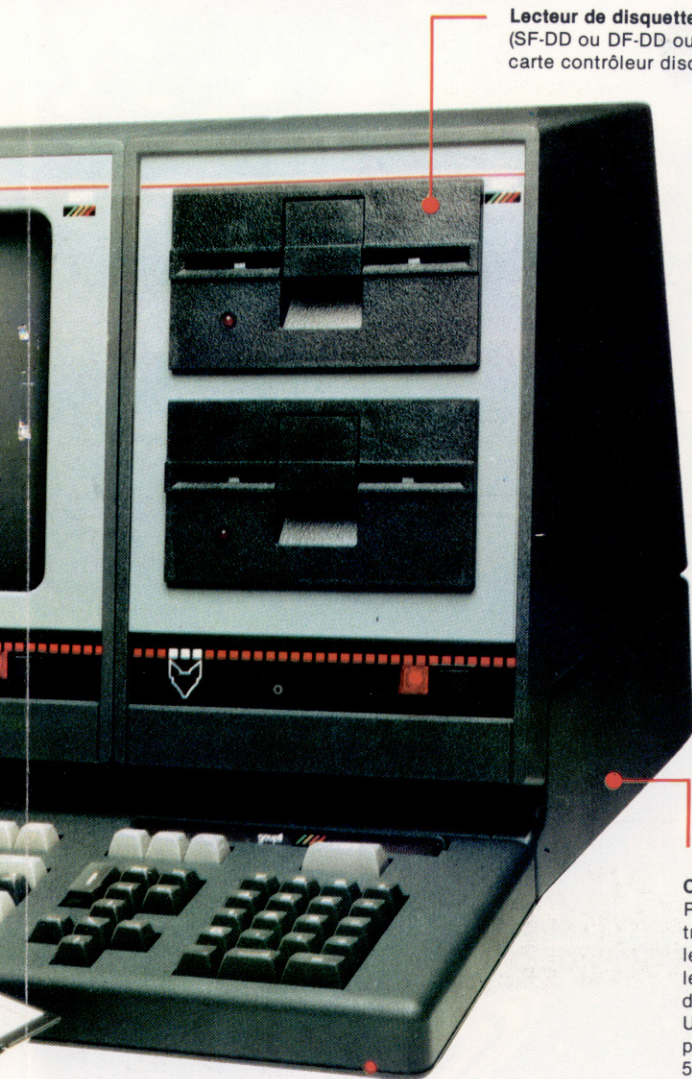
Sortie série

Sortie vidéo moniteur N/3

**Sortie vidéo couleurs**

(moniteur ou téléviseur sur prise Péritel)

D'autres cartes sont présentes dans le Goupil 3, telle la carte contrôleur de lecteurs de disquettes 5 pouces gérant les unités de disquettes.



Lecteur de disquette 5 pouces  
(SF-DD ou DF-DD ou DD-DT) avec  
carte contrôleur disquette.

#### Console

Possibilité de choisir entre  
trois microprocesseurs différents,  
le 6809 2 MHz, le Z80 4 MHz ou  
le 8088 16 bits 5 MHz. 64 ou 128 K  
de mémoire RAM (extensible à 1 Mo).  
Une interface série, deux interfaces  
parallèles. Boîtier alimentation  
50 ou 100 W.

## GOUPIL

### PRIX

\*\*\*\*

### DIMENSIONS

Clavier :  
525 x 185 x 65 mm.  
Console :  
525 x 340 x 125 mm.

### UC

6809 Motorola, Z80 Zilog  
ou 8088 Intel.

### MÉMOIRE

64 K en RAM extensible  
à 1 Moctets.

### ÉCRAN

25 lignes de 80 caractères,  
affichage texte aux normes  
Vidéotex.

### INTERFACES

On retiendra les cartes :  
extension entrées/sorties,  
Vidéotex, graphique  
couleurs, entrées/sorties  
série, IEEE-488, etc.

### LANGAGE

Voir encadré.

### CLAVIER

101 touches réparties  
sur trois blocs, plus une  
rangée supérieure de  
touches de fonction.

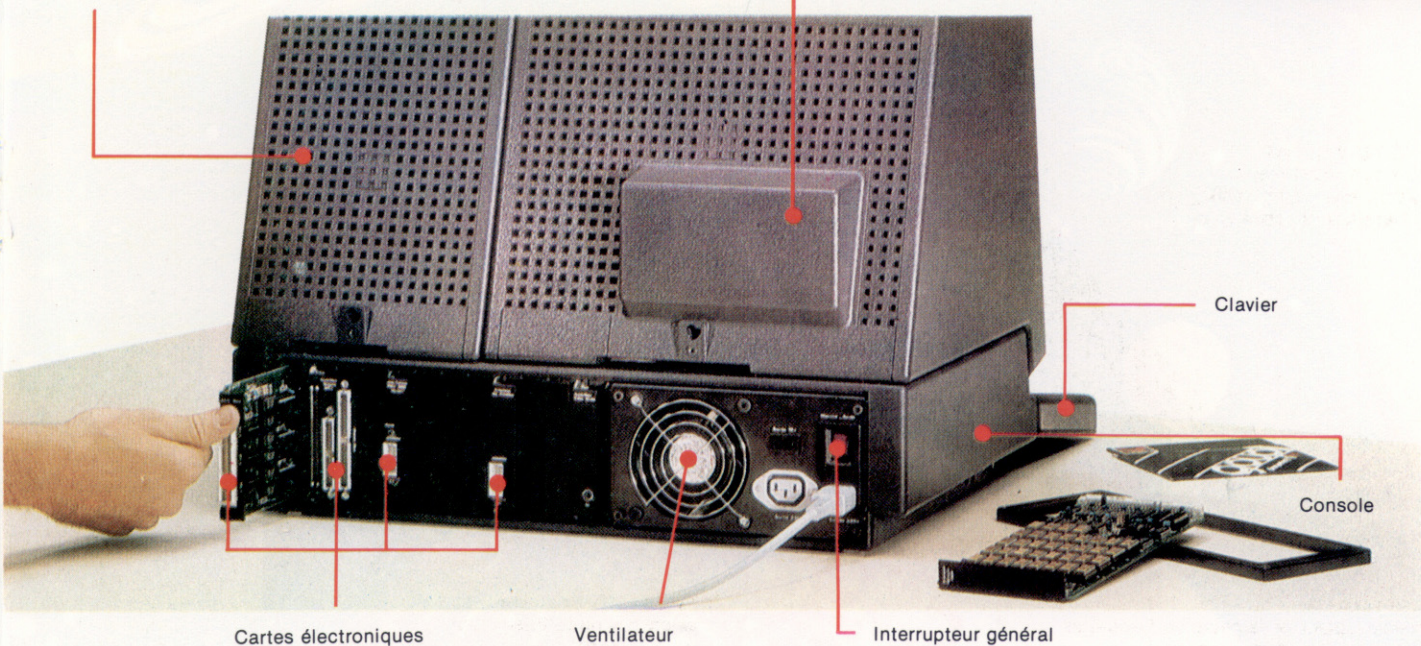
### DOCUMENTATION

Très détaillée mais pas  
très simple.

Lecteur de disquette

Face arrière de Goupil

Vidéo 25 x 80



Cartes électroniques

Ventilateur

Interrupteur général

Clavier

Console

# Atterrissage

**Lunar Lander est un jeu qui peut sembler trop simple en comparaison de certains jeux d'arcades très rapides et très bruyants. Mais correctement programmé, il devient subtil et difficile à maîtriser.**

## Attraction planétaire

Ces valeurs donnent l'attraction gravitationnelle approximative du Soleil, de la Lune et des planètes de notre système solaire en mètres par seconde carrée. Ces valeurs (négatives pour le jeu), pour la variable  $g$  du programme, doivent être entrées à la ligne 20. Il existe évidemment des corps sur lesquels il serait suicidaire d'atterrir (comme le Soleil ou Jupiter) mais ce problème peut être négligé ici. (Cl. Adrian Morgan.)

Dans Lunar Lander, vous devez poser un vaisseau sur la Lune (ou sur une autre planète). L'ordinateur de bord étant en panne, vous devez poser le vaisseau délicatement à l'aide de courtes mises à feu des réacteurs. Il s'agit de heurter la surface de la planète à une vitesse raisonnable et de trouver le juste milieu entre une descente trop rapide et l'épuisement d'une quantité limitée de carburant. Ce jeu ressemble à une simulation. L'idée est de refléter le comportement réel d'un vaisseau spatial et cela aussi précisément que possible.

Examinons de plus près les problèmes posés par un atterrissage :

— La planète présente une certaine attraction. Elle est la cause d'une accélération du vaisseau au cours de sa descente vers la surface de la planète.

— Le vaisseau est muni d'un réacteur qui contre les effets de la gravité en poussant le vaisseau vers le haut.

— Le vaisseau a une masse (un poids). Plus celle-ci est élevée, moins grande est l'efficacité du réacteur. Le poids du vaisseau est composé de son propre poids plus celui de son carburant. La consommation de ce carburant entraîne une diminution du poids du vaisseau.

Ainsi, pour simuler l'atterrissage, nous avons besoin d'une série d'équations impliquant l'accélération, la masse, la vitesse, etc. Ce peut être très compliqué ou très simple, selon le détail et la précision recherchés. Dans le cas de ce jeu, nous avons décidé d'opter pour la simplicité.

Le principal paramètre est l'altitude du vaisseau. Évidemment, elle varie constamment. Pour nous permettre de déterminer sa position à tout moment, nous divisons le « temps » en une série de périodes.





Dans chaque période, nous pouvons calculer la distance parcourue par le vaisseau, le changement survenu dans sa masse et dans sa vitesse et ainsi de suite. Vous pouvez choisir n'importe quelle longueur de période — plus la période est courte, meilleure sera la simulation. L'introduction de la notion de périodes facilite l'écriture des équations.

La vitesse est mesurée en unités de distance par heure. En deux heures, une bicyclette se déplaçant à 10 km/h aura parcouru 20 km. En trois heures, elle aura franchi 30 km.

$$\text{Distance} = \text{temps} \times \text{vitesse.}$$

Nous pouvons donc calculer la distance (vers le haut ou vers le bas) parcourue par le vaisseau en multipliant sa vitesse par la longueur de la période (que nous définirons comme étant une unité). Nous pouvons modifier la vitesse au moyen de l'accélération due à l'attraction gravitationnelle ou de la décélération due à la poussée des réacteurs.

L'accélération due à la gravité est toujours constante (la variable *g* dans le programme) et dépendra de la planète approchée. L'illustration donne les valeurs pour les planètes de notre système solaire, mais vous pouvez faire l'essai d'autres valeurs ou faire en sorte que le programme génère une valeur aléatoire pour *g*, pour rendre le jeu plus difficile.

La simulation du réacteur est un peu plus compliquée. Dans cette version, le joueur peut brûler jusqu'à 9 unités de carburant dans une période donnée, et le programme calcule l'accélération résultante, compte tenu de la masse du vaisseau. La formule exacte dépend de la puissance du réacteur et du type de carburant utilisés. Dans ce programme, nous avons choisi des valeurs donnant un jeu difficile et intéressant; vous pouvez essayer de les modifier pour varier le déroulement du jeu.

Par ailleurs, la simulation doit être faite en temps réel. On a souvent abusé de cette expression; elle signifie que le programme (jeu, simulation, etc.) tourne continuellement, ne laissant jamais au joueur la possibilité d'entrer des données ou des commandes. C'est souvent la seule différence existant entre l'utilisation d'une commande INPUT pour entrer des données et l'emploi d'une commande INKEY\$ ou GET.

Le jeu Lunar Lander sera évidemment de meilleure qualité si le programme ne s'arrête pas pour calculer la réserve de carburant. S'il s'arrêtait, le joueur aurait le temps d'étudier la situation. Dans la réalité, il devrait plutôt faire preuve de vivacité d'esprit.

Dans notre programme d'atterrissage, une boucle est exécutée à chaque intervalle de temps. En choisissant une période, qui est en fait égale au temps d'exécution de la boucle, la simulation fonctionne en temps réel. Si l'atterrissage simulé peut être aussi long que dans la réalité, il est plus court dans ce jeu pour ne pas devenir fastidieux.

Vous pouvez apporter de nombreuses améliorations au programme de base. La plus évidente

serait d'ajouter un affichage graphique représentant la descente. Ce pourrait être un altimètre, une vue latérale du vaisseau, ou même une représentation à l'échelle de la piste d'atterrissage avec illustration de l'approche. Vous pouvez aussi ajouter un déplacement latéral afin de devoir positionner le vaisseau sur la piste. Le vaisseau ne sera pas dévié latéralement dans l'espace, puisqu'il n'existe aucune force pour le pousser ou pour le tirer dans une autre direction que celle du sol. Mais si vous vous posez sur une planète ayant une atmosphère, vous pouvez tenir compte du problème des vents de surface. Certaines versions sophistiquées du programme comportent plusieurs zones d'atterrissage situées près de tunnels et de cratères, de telle sorte qu'il faut manœuvrer avec beaucoup de précision.

Lunar Lander, comparé aux jeux d'arcades modernes, peut sembler démodé. Mais sa programmation et son utilisation peuvent constituer pour de nombreuses personnes une première expérience intéressante de simulation sur ordinateur. L'écriture de ce programme peut ouvrir la porte à une toute nouvelle gamme de projets de programmation.

Or l'expérience prouve que de nombreux informaticiens amateurs ont eu de bonnes idées de jeux qu'ils n'ont pas toujours su développer et rédiger. Il faut donc commencer par les exercices de programmation les plus simples avant de se lancer dans la rédaction d'un programme de jeux qui soit vraiment compétitif avec ceux qui existent déjà sur le marché.

### Atterrissage

```

10 REM Jeu Lunar Lander
20 LET g = 1.6
30 LET t = 1
40 LET c = 1000
50 LET v = 0
60 LET h = 2000
70 LET m = 2000+c
80 LET g = g*t
100 REM mise à jour de l'écran
110 PRINT AT 0,0
120 PRINT "          Lunar Lander"
130 PRINT: PRINT "Altitude ....":INT h: " "
140 PRINT: PRINT "Vitesse ....":INT v: " "
150 PRINT: PRINT "Carburant ....":INT c: " "
160 PRINT
165 IF h < 0 THEN GO TO 400
170 IF t < 0 THEN LET t = 0: PRINT "***
Carburant épuisé ***": GO TO 190
180 PRINT "Mise à feu du réacteur 0-9"
190 LET b = 0: IF c > 0 THEN LET a$=INKEY$:IF
a$ < ">" THEN LET b=VAL a$
200 IF b > c THEN LET b=0
210 LET h = h+v*t
220 LET v = v+g
230 LET v = v+(b*3000)/m
240 LET c = c-b: LET m=m-b
250 FOR i = 1 TO 50: NEXT i
300 GOTO 110
400 REM sur la surface de la planète
410 IF v > -10 THEN PRINT "Atterrissage réussi": GO TO 500
420 IF v > -20 THEN PRINT "*** Ecrasement! Vous
avez abîmé le vaisseau mais l'équipage n'a rien!" :GO TO 500
430 PRINT "Ecrasement fatal, vaisseau détruit,
aucun survivant"
440 PRINT: PRINT "Vous venez de créer un
cratère dont le diamètre fait ": INT (-v*2.1):" kilomètres"
500 PRINT: PRINT "un autre jeu
(O/N)?" :
510 LET a$ = INKEY$: IF a$ = "" THEN GO TO 510
520 IF a$="o" OR a$="O" THEN RUN
530 IF a$ < ">" AND a$ < ">" THEN GO TO 510
    
```

### Variantes de basic

Voici un listing de Spectrum; les autres machines n'exigent pas l'utilisation du mot LET. Sur le BBC, remplacez la ligne 10 par :

```
10 PRINT TAB(0,0)
```

Remplacez INKEY\$ dans les lignes 190 et 510 par INKEY\$(0). Remplacez VAL a\$ de la ligne 190 par VAL(a\$). Remplacez INT h et INT v dans les lignes 130 et 140 par INT(h) et par INT(v).

Sur le Commodore 64 et sur le Vic-20, remplacez la ligne 110 par :

```
110 PRINT CHR$(19)
```

Remplacez LET a\$=INKEY\$ dans les lignes 190 et 510 par GET a\$. Remplacez VAL a\$ de la ligne 190 par VAL(a\$). Remplacez INT h et INT v dans les lignes 130 et 140 par INT(h) et par INT(v).

Sur l'Oric Atmos, remplacez la ligne 110 par :

```
110 PRINT 0,0
```

Remplacez INKEY\$ dans les lignes 190 et 510 par KEYS. Remplacez VAL a\$ de la ligne 190 par VAL(a\$). Remplacez INT h et INT v dans les lignes 130 et 140 par INT(h) et par INT(v).

# Question de style

**Documenter un programme recouvre bien d'autres aspects que de lui adjoindre des commentaires. Voyons comment un programme en basic ou en pascal peut être documenté de manière satisfaisante.**

Prenez la première version de notre programme (listage 1). C'est manifestement une énigme : que fait-il? Après avoir vu qu'il entrait deux nombres, les multipliait chacun par un nombre et ajoutait les résultats entre eux pour imprimer la solution, on ne voit pas ce que le code réalise. Examinez maintenant la deuxième version du programme (listage 2) : tout devient clair. Et pourtant il n'y a pas eu de commentaires ajoutés, ni de titre au programme ni de lignes de REMarques. En outre, aucune documentation externe n'a été fournie.

Il est intéressant d'étudier en détail les différences entre ces deux versions. D'abord les nombres qui figuraient au premier listage ont été remplacés par des noms (AANNEE et AMOIS). Les nombres qui ne changent pas lors de l'exécution du programme sont appelés des *constantes*. Certains langages, tel le PASCAL, ont adopté une notation spéciale pour les constantes (au listage 2, les deux constantes sont définies différemment des variables). D'autres langages, comme le BASIC, n'ont pas de notation particulière pour les constantes (les lignes 10 et 20 du programme BASIC utilisent des noms de variables pour définir les constantes). Donner des noms aux constantes n'est approprié que si elles doivent être utilisées de manière fréquente. Si ce n'est pas le cas, des commentaires insérés dans le programme conviennent mieux.

La deuxième amélioration fondamentale est que les noms de variables prêtant à confusion ont été remplacés par d'autres plus longs et intelligibles. Ceux que nous avons choisi ici (NANNEES pour remplacer A, agensecs au lieu de e, etc.), ont été retenus parce qu'ils font moins de 10 caractères et que leurs deux premiers caractères peuvent les différencier.

Généralement on donne aux variables des noms en rapport avec leur rôle dans le programme. Ainsi vous appellerez un compteur ordinal d'itération BOUCLE (au lieu des noms de variables habituels I et J). La première et la dernière valeur prises par le compteur peuvent être affectées à des constantes ou à des variables aux noms déterminés. Ainsi une boucle :

```
FOR J = 1 TO 10...NEXT J
```

deviendrait :

```
FOR BOUCLE = PREMIERE TO DIXIEME...NEXT BOUCLE
```

Des noms de variables plus longs prendront bien sûr plus de temps à taper et davantage de place mémoire. Mais ils présentent l'avantage

de rendre les programmes plus intelligibles et de faciliter la recherche d'erreurs. Si votre langage n'utilise que les deux premiers caractères des noms pour les différencier, vous devrez vous assurer que les noms que vous choisissez diffèrent bien par leurs deux premiers caractères. Ainsi deux noms longs de variables (par exemple CODENOM et COMP) peuvent sembler différents sans pour autant être différenciés par le programme.

Une autre différence fondamentale entre les listages est que le deuxième utilise des messages guide-opérateurs longs et significatifs pour les entrées, et donne des explications utiles en sortie (lignes PRINT en BASIC, « write » en PASCAL). Cela permet deux choses : d'abord un programme plus lisible, même si les variables étaient de simples lettres ; ensuite, et c'est le plus important, le programme devient accessible même pour qui ne l'aurait pas vu auparavant.

## Présentation du programme

Les utilisateurs du langage PASCAL savent déjà l'importance de bien présenter à l'écran les programmes. Respecter de simples conventions, comme faire des alinéas, ménager des lignes blanches et utiliser de manière appropriée des majuscules et des minuscules, peut transformer une masse informe de symboles en un exposé logique et lisible. Le formatage d'un programme pour l'imprimante ou pour l'écran est facilité par une construction syntaxique sous forme de boucles (FOR...NEXT, WHILE...WEND, REPEAT...UNTIL), et par des boucles imbriquées les unes dans les autres.

Cela dit, il est regrettable que la plupart des BASIC laissent si peu de choix quant à la présentation d'un programme. Les langages compilés, tels que le PASCAL, sont bien plus souples dans la mesure où ils utilisent un éditeur de texte (ou un traitement de texte). Par ailleurs, l'édition d'un programme BASIC est souvent assez difficile (à moins que, comme le MBASIC de Microsoft, votre interpréteur accepte une version ASCII du programme et en fasse un élément transformable en un programme exécutable). Pire encore, de nombreux BASIC reformateront vos programmes en supprimant les alinéas ! D'autres, par contre, se chargeront d'en créer, le BBC Micro, par exemple, à condition néanmoins que vous lui ayez spécifié la commande LISTO. La plupart des systèmes PASCAL comprennent un formateur qui est d'une grande utilité.



Il est de toute manière nécessaire que vous mettiez au point vos propres conventions d'écriture.

Les commentaires sont bien sûr le meilleur moyen pour documenter vos programmes de manière interne. Les conventions varient également d'un langage à l'autre. Le BASIC utilise l'instruction REM, laquelle doit figurer devant votre commentaire. L'interpréteur ignore alors tout ce qui le suit, jusqu'au prochain marqueur de fin d'instruction (ou : retour chariot). Dans d'autres langages (PASCAL, PL/I, PROLOG, etc.), les commentaires sont mis entre parenthèses par les signes /\* et \*/ (quelquefois des accolades). Tout ce qui figure alors entre les parenthèses est ignoré par l'interpréteur. Ce système présente l'avantage de permettre aux commentaires de figurer sur plusieurs lignes, l'inconvénient étant que, si vous oubliez le deuxième \*/, le reste du programme est considéré comme un commentaire et ignoré!

Vous introduirez des commentaires chaque fois que vous ressentirez le besoin de fournir des explications : lorsque vous définissez des constantes, initialisez des variables, au début d'un programme, d'une nouvelle procédure (sous-programme), lors de la définition d'une fonction ou de l'écriture d'un code peu lisible du fait de sa complexité.

Les commentaires ne doivent pas nécessairement être longs et explicites, un mot peut suffire pourvu qu'il soit significatif et vous permette de vous rappeler de quoi il s'agit. Ainsi, lorsque vous essayez de vous remémorer la logique d'un programme de jeu fait l'année précédente, de gros pavés de commentaires qui interrompent le code sans pour autant donner de détails sont davantage des obstacles à la compréhension du programme que des aides. Les commentaires doivent être succincts et opportuns. Insérez-en avant les parties du programme dont le code est embrouillé et évitez d'en mettre au beau milieu du raisonnement. La présence des commentaires ne devrait jamais venir gêner la lecture et la compréhension de la structure du programme. La version finale de notre programme (listage 3) donne quelques exemples et principes directeurs.

La documentation externe, sous la forme de manuels et de spécifications écrites, est la partie la plus difficile à réaliser. Pour les programmeurs, des études ont confirmé le fait que les documentations ne sont consultées qu'en dernier recours. Il ne faut pourtant pas oublier qu'elles peuvent faire gagner beaucoup de temps. Si votre programme n'est pas trop long et bien documenté de manière interne, il y a peu de chance que vous ayez besoin d'une documentation séparée. La documentation de l'utilisateur est une question à part et sera abordée plus tard. Même si vous n'avez pas besoin d'écrire une documentation séparée pour vos programmes, vous ressentirez le besoin d'avoir une documentation de référence lorsque vous voudrez reprendre un vieux programme ou dépister les erreurs d'un nouveau.

## Code documenté de manière satisfaisante

### Listage 1 BASIC

```
(a) 10 INPUT A,B
    20 C=A*31536000
    30 D=B*2592000
    40 E=C+D
    50 PRINT E
```

### PASCAL

```
(b) localisation du programme (entrée,sortie):
var a,b,c,d,e: valeur entière
begin
read(a,b);
c:=a*31536000;
d:=b*2592000;
e:=c+d;
writein(e);
end.
```

### Listage 2 BASIC

```
(a) 10 AANNEE=31536000
    20 AMOIS=2592000
    30 PRINT"SAISIR VOTRE AGE (les années puis les mois, séparés par une virgule) ";
    40 PRINT NANNEES,NMOIS
    50 YSECS=NANNEES*AANNEE
    60 MSECS=NMOIS*AMOIS
    70 AGENSECS=YSECS+MSECS
    80 PRINT"Votre âge en secondes est (approximativement) "IAGENSECS
```

### PASCAL

```
(b) Programme asensecondes (entrée,sortie):
const
aannées=31536000;
amois=2592000;
var
nannées,nmois,ysecs,msecs,asensecs:valeur
entière;
begin
write("saisissez votre âge (années puis mois séparés par une virgule) ");
read (nannées,nmois);
ysecs:=nannées*aannée;
msecs:=nmois*amois;
asensecs:=ysecs+msecs;
writein("votre âge en secondes est (approximativement) ",asensecs); end.
```

### Listage 3 BASIC

```
(a) 10 REM "AGENSECONDES" JUIN 1984
    20 REM saisir par INPUT l'âge en années et en mois (y,m) et
    30 REM utilise une conversion approximative (1 mois = 30 jours)
    40 REM pour donner l'âge en secondes
    50 REM
    60 AANNEES=31536000:REM secondes en 365 jours
    70 AMOIS=2592000:REM secondes en 30 jours
    80 PRINT"Entrez votre âge (années puis mois séparés par des virgules) ";
    90 INPUT NANNEES,NMOIS
    100 REM l'âge en secondes est (âge en années * secondes d'une année) plus (mois
    depuis l'anniversaire * secondes d'un mois)
    120 MSECS=NMOIS*AMOIS
    130 AGENSECS=YSECS+MSECS
    140 PRINT"votre âge en secondes est (approximativement) "IAGENSECS
```

### PASCAL

```
(b) programme asensecondes (entrée,sortie):
/* Juin 1984
Lit l'âge en années et en mois (y,m) et utilise une conversion approximative (1
mois = 30 jours), pour donner l'âge en secondes. */
const
aannée 31536000; /* secondes dans 365 jours */
amois 2592000; /* secondes dans 30 jours */
var
nannées, mois, ysecs, msecs, asensecs:valeur entière;
begin
séparés d'une virgule) ";);
read(nannées,nmois);
/* l'âge en secondes est (âge en années*secondes d'une année) plus
(nombre de mois depuis le dernier anniversaire * secondes d'un mois) */
ysecs:=nannées*aannée;
msecs:=nmois*amois;
asensecs:=ysecs+msecs;
writein("votre âge en secondes est (approximativement) ",asensecs);
end.
```



# Le nucléaire en famille!

« Apocalypse » est un jeu classique, de type Diplomacy, qui nécessite une habileté tactique certaine et des réactions rapides. Mais attention, vous risquez de déclencher... l'apocalypse.

La plupart des jeux pour ordinateurs ont quelque chose d'anti-social. La publicité peut bien nous présenter une famille regroupée autour de la fabuleuse machine, ordinairement, un seul joueur à la fois en fait usage.

De ce point de vue, « Apocalypse » est propice aux réunions de famille et recrée l'atmosphère de rivalité et d'intrigue typiques des jeux de société comme « Diplomacy ». La version destinée au BBC Micro peut même accepter jusqu'à quinze joueurs, et ils ne se contentent pas de prendre leur tour l'un après l'autre : c'est un vrai jeu collectif.

D'ailleurs Apocalypse était originellement un authentique jeu de société; il a simplement été adapté pour ordinateur. Le jeu proprement dit se déploie sur quatre théâtres d'opérations : l'Europe, les Caraïbes, le Royaume-Uni et Londres. De surcroît, il est possible d'acquérir des extensions, la guerre prenant place cette fois en Afrique du Sud, au Moyen-Orient, dans l'Arctique, aux États-Unis, en Asie du Sud-Est; d'autres recréent la guerre du Pacifique, la chute de l'Empire romain ou les batailles de Napoléon. Toutes ces options sont disponibles sur cassette et sont fusionnées avec le programme principal.

L'écran est divisé en carrés : 20 x 20 pour le Spectrum, 40 x 40 pour le BBC Micro. Le bleu représente évidemment la mer, et les autres couleurs diverses particularités géographiques : zones urbaines ou rurales, villes, montagnes, déserts. Des signes conventionnels permettent d'identifier les forces en présence. La qualité graphique est bien sûr fonction des possibilités de l'appareil utilisé. Sur le Spectrum, les carrés

sont ou pleins ou vides, et il est facile de voir où l'on en est exactement. Le BBC Micro offre une représentation plus détaillée, mais l'écran semble parfois un peu encombré; les forces en présence se distinguent mal sur l'arrière-plan.

Les joueurs déploient leurs troupes chacun à leur tour, selon le scénario choisi. Puis, la confrontation étant bien en place, les attaques ont lieu et l'ordinateur sert d'arbitre. Les différents mouvements envisageables sont sélectionnés à partir de plusieurs menus; de ce point de vue, il ne fait aucun doute que l'affichage pourrait être plus compréhensible. La guerre nucléaire est en fait une option parmi d'autres, et y recourir entraîne des résultats prévisibles... A noter qu'il faut respecter la vérité historique : il n'est donc pas possible de lancer une ogive nucléaire sur des légions romaines hostiles!

Le jeu est relativement lent, mais cela peut être un atout pour certains amateurs de ce genre de jeux. Bien entendu, il n'est pas forcément nécessaire de jouer avec l'ordinateur; Apocalypse demeure aussi passionnant, joué à plusieurs autour d'une table. Il s'agit d'une bonne introduction aux jeux plus complexes développés pour Apple et Commodore.

## Apocalypse :

pour le BBC Micro (de 2 à 15 joueurs);  
pour le Spectrum (de 2 à 4 joueurs).

Éditeur : Red Shift.

Auteurs :

Helmut Watson (BBC), Bob Tyler (Spectrum).

Créateur du jeu d'origine : Mike Hayes.

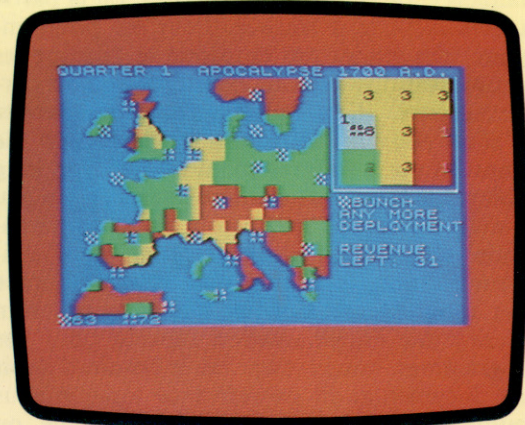
Extensions : voir liste dans le corps de l'article.

Manches à balai : inutiles.

Format : cassette.

## La guerre à l'écran

Apocalypse est un jeu typiquement familial : faisant intervenir de 2 jusqu'à 15 joueurs dans la version BBC, jusqu'à 4 dans la version Spectrum présentée ici. Les adversaires se livrent une lutte armée qui peut parfois déboucher sur un conflit nucléaire. Fort heureusement, l'écran se limite à des tableaux de ce genre, et nous épargne les images de la désolation.  
(Cl. Liz Heaney.)





# Diriger l'action

**Les bons jeux d'arcades doivent être écrits en langage machine. C'est une vraie gageure pour un débutant. Aussi présentons-nous ici une routine de lutins en langage machine pour le Spectrum.**

Il n'est pas facile d'écrire des routines graphiques en langage d'assemblage, mais le programme présenté ici devrait vous donner une bonne idée de l'approche de cette tâche. Le programme imprime un arrière-plan constellé d'astérisques, et vous permet de déplacer une forme définie par l'utilisateur (dans notre exemple, une croix) sur tout l'écran en appuyant sur les touches de curseur. La croix se déplace sans à-coups, avec des pas d'un pixel vers le haut, le bas, la droite ou la gauche, et laisse l'arrière-plan inchangé. La routine de mouvement du lutin peut facilement être incorporée dans vos programmes BASIC.

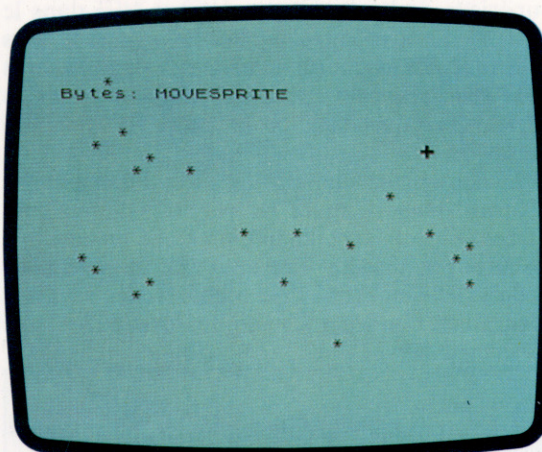
Pour entrer le programme dans votre Spectrum, il vous faut d'abord taper le programme BASIC. Le langage machine peut ensuite être entré, soit en le tapant dans le programme de chargement BASIC, qui lit le code à partir des instructions data, puis les entre en mémoire par POKE, soit en entrant le langage d'assemblage source au moyen d'un assembleur adéquat. Les deux types de programmes peuvent être sauvegardés sur cassettes aux lignes 9000 et 9010 du programme BASIC.

Pour comprendre comment marche le programme, commençons par regarder le programme BASIC. Le sous-programme à la ligne 1000 lit la définition du lutin à partir des instructions data et l'écrit (POKE) en RAM où elle peut être utilisée par le programme en langage machine. Les lignes 90 à 110 impriment l'arrière-plan, et la ligne 120 fixe la position de départ de la croix. PRINT AT 10,16 a pour effet que l'interpréteur BASIC calcule l'adresse écran correspondant à ces coordonnées, et cette adresse est stockée dans la variable système DFCC (adresses 23684 et 23685) où elle peut être lue par le programme en langage machine. La ligne 130 appelle la section d'initialisation du programme en langage machine. Les lignes 140 à 180 sont une boucle qui attend qu'une touche soit enfoncée, qui écrit (POKE) la valeur de la touche dans un emplacement mémoire à lire par le langage machine, puis appelle ce dernier pour déplacer le lutin d'un pixel dans la direction indiquée par la touche.

Le programme en langage d'assemblage commence par définir des noms pour les emplacements mémoire utilisés. KEY est l'emplacement où la valeur de la touche est stockée. SPRPOS sert à garder l'adresse mémoire de la position d'écran où le lutin apparaîtra. SPRTAB est un tableau dans lequel le programme stocke la défini-

tion du lutin et le contenu des positions d'écran qui peuvent être occupées par le lutin. Celui-ci peut se déplacer n'importe où dans l'écran, pas seulement en sauts de rectangles de caractères entiers. Ainsi les huit bits de chaque rangée du lutin peuvent être divisés en deux octets de mémoire écran, et le tableau utilise deux octets pour stocker les huit bits, partagés de la même façon que sur l'écran. L'emplacement mémoire BITPOS est utilisé pour stocker le nombre de bits dont la donnée du lutin a été déplacée depuis le début de l'octet.

La partie initialisation du programme lit l'adresse écran de départ du lutin à partir de DFCC, puis saute à la section étiquetée SAVSCR où l'on stocke l'adresse écran en SPRPOS, charge la



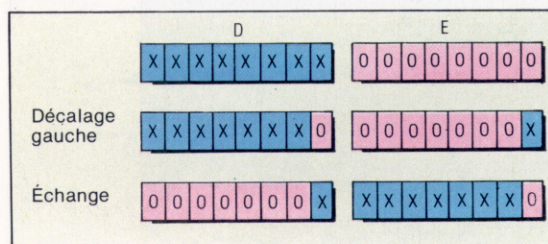
## Lutin en mouvement

Le programme de démonstration basic déplace le lutin (initialisé par une croix dans les instructions data) à travers un fond d'étoiles, en réponse aux touches de curseur.

(Cl. Liz Heaney.)

valeur 1 dans le registre D, et appelle le sous-programme UNDER. Lorsque D est mis à 1, UNDER copie le contenu de la zone écran dans laquelle le lutin sera imprimé, de sorte que l'arrière-plan peut être remis à l'état initial une fois que le lutin s'est déplacé. Le programme appelle ensuite le sous-programme PRSPRT pour imprimer le lutin à l'écran.

La partie du programme qui s'occupe du mouvement du lutin commence au label MOVSPR. Elle commence par charger 0 dans le registre D et appeler le sous-programme UNDER. Avec D à



## Décalage et échange

Le décalage de chacun des octets du lutin (représentés ici par des X) dans le registre DE pour effectuer un mouvement d'écran à gauche ou à droite peut « enrôler » un bit. Si cela arrive, D et E sont échangés, ce qui réunit les bits du lutin.

(Cl. Kevin Jones.)

**Adressage d'écran**

L'écran à 24 lignes du Spectrum est divisé en trois sections de huit lignes. Chaque ligne d'écran est divisée en huit rangées haute résolution (HIRES) de 32 octets, dont chaque bit correspond à un pixel. Les octets sont numérotés séquentiellement le long d'une rangée hires, et de chaque rangée dans une ligne d'écran à la rangée correspondante de la ligne d'écran d'en dessous; ainsi, les octets des huit rangées supérieures des huit lignes d'une section d'écran ont des adresses consécutives. L'adresse suivante est celle du premier octet de la deuxième rangée de la première ligne d'écran; tous les octets des huit secondes rangées hires de cette section d'écran sont adressés séquentiellement à partir de cette adresse, et ainsi de suite. Le programme suivant pour le Spectrum 48 K illustre cela :

```
50 LET SCRSTART=16384
60 LET SCRNEND=22527
100 FOR B=SCRSTART TO
   SCRNEND
200 POKE B,255
300 NEXT B
```

zéro, UNDER recopie sur l'écran le contenu d'écran préalablement sauvegardé dans le tableau, en effaçant le lutin de l'écran. Le programme lit ensuite la position du lutin et la valeur de la touche, teste cette dernière, appelle la routine pour préparer le mouvement dans la direction appropriée, puis saute en SAVSCR pour sauvegarder le contenu d'écran et imprime le lutin sur l'écran, exactement comme précédemment.

Pour comprendre les deux routines ABOVE et BELOW qui préparent les mouvements vers le haut et le bas du lutin, il nous faut d'abord examiner la façon étrange dont le Spectrum fait correspondre des adresses mémoire aux positions d'écran. Tout cela est expliqué au chapitre 24 du manuel du Spectrum. Si l'on regarde les adresses en hexadécimal, on voit que, pour les 256 caractères de chaque section de l'écran, l'octet le moins significatif de l'adresse (lo) de chacun des huit octets qui forment chaque caractère est le même que le nombre de caractère dans le bloc, alors que l'octet le plus significatif de l'adresse (hi) est augmenté de un lorsque nous déplaçons une ligne de pixels vers le bas de l'écran. Pour cette raison, les huit rangées de pixels pour un caractère ont leurs adresses comprises entre \$4000 et \$47FF dans le tiers supérieur de l'écran, entre \$4800 et \$4FFF dans le tiers médian, et entre \$5000 et \$57FF dans le tiers inférieur. Le signe (\$) signifie que les nombres sont en notation hexadécimale — certains assembleurs utilisent à la place le symbole dièse (#).

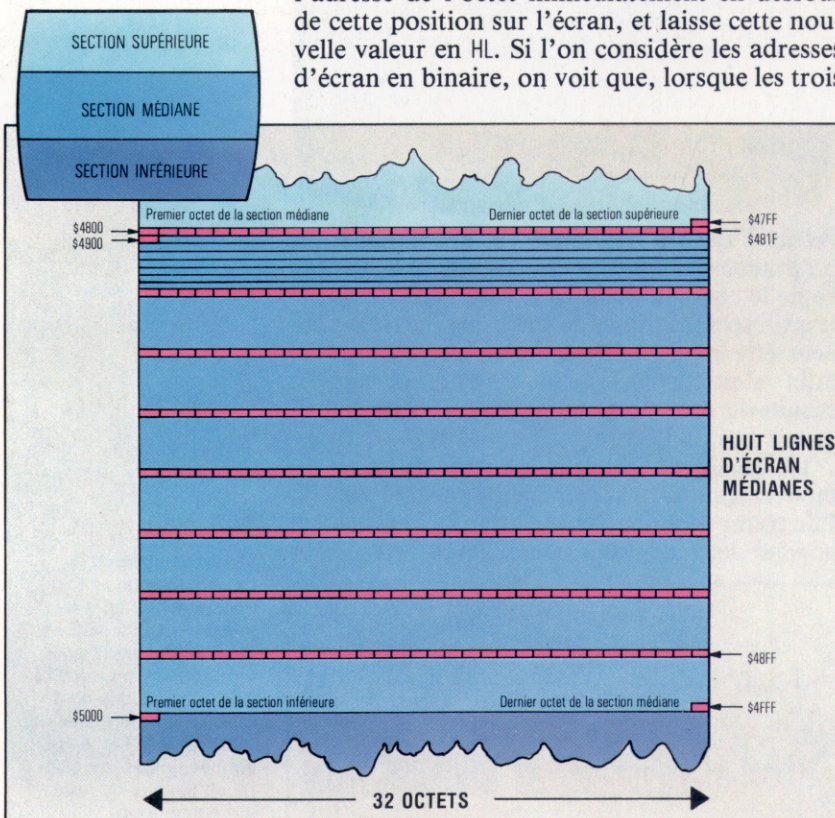
Le sous-programme BELOW attend une adresse d'écran dans la paire de registres HL, calcule l'adresse de l'octet immédiatement en dessous de cette position sur l'écran, et laisse cette nouvelle valeur en HL. Si l'on considère les adresses d'écran en binaire, on voit que, lorsque les trois

bits inférieurs de H sont 111, la ligne suivante de pixels est dans un autre bloc de caractères. BELOW teste d'abord cela; si nous sommes toujours dans le même bloc de caractères, tout ce qu'il faut faire est d'ajouter 1 à H. Si nous sommes dans un autre bloc, il faut ajouter \$20 (puisque'il y a 31 caractères dans une ligne) à L. Si la nouvelle valeur de L est comprise entre 0 et \$1F (les trois bits supérieurs 000), cela signifie que nous sommes dans une autre section d'écran. La valeur de HL est l'adresse écran en cours.

Si nous sommes toujours dans le même bloc écran, il faut soustraire 7 de H. Cela vous paraîtra plus facile à comprendre si vous travaillez en code et voyez ce que cela fait à l'adresse montrée dans le tableau. ABOVE est semblable à BELOW, mais calcule l'adresse du pixel au-dessus de la position d'écran.

Les sous-programmes LMOVE et RMOVE déplacent la configuration des bits de pixels dans le tableau, respectivement à gauche et à droite. Encore une fois, ils sont très semblables, et nous examinerons donc seulement LMOVE. L'accumulateur est chargé avec le pointeur de position de bit, qui est un nombre à un octet compris entre 0 et 7, correspondant à la numération des bits dans un octet. On soustrait ensuite 1 de la valeur de l'accumulateur pour effectuer le mouvement; il en résultera aussi un nombre compris entre 0 et 7, à moins que la valeur initiale du pointeur de position de bit soit 0, auquel cas l'accumulateur contiendra 255. L'utilisation de l'instruction AND 7 assure que la valeur dans l'accumulateur demeure comprise entre 0 et 7. Nous avons alors une boucle pour les huit lignes de pixels dans le lutin. Pour chaque ligne, on charge deux octets du tableau contenant cette ligne de pixels dans la paire de registres DE et on effectue une permutation gauche de 16 bits sur DE. Si cela ne fait pas passer un bit du lutin de l'extrémité supérieure de D vers l'extrémité inférieure de E, on peut simplement stocker la configuration de lutin décalée à nouveau dans le tableau et passer à la ligne de pixels suivante. Si l'on a déplacé un bit du lutin de D à E, il faut échanger D et E avant de les remettre dans le tableau. A la fin de cette routine, il faut aussi soustraire 1 de HL afin que le lutin soit imprimé à gauche.

Le sous-programme final du programme est PRSPRT, qui effectue l'impression du lutin sur l'écran. Il consiste en deux boucles emboîtées, l'une pour les huit lignes de pixels du lutin, contrôlée par le registre C, l'autre pour les deux octets entre lesquels est partagée la ligne de pixels, contrôlée par le registre B. La partie importante de la routine est la section centrale qui stocke les bits de la configuration du lutin. Nous avons l'adresse écran dans la paire de registres HL et l'adresse du tableau de lutin dans le registre IX. PRSPRT prend un octet de la configuration de pixels de l'écran, et l'associe par OU (OR) avec ce qui est déjà à l'écran, de sorte que nous terminons avec les points noirs provenant de la configuration de pixels du lutin surimposés au contenu précédent de l'écran.





## Mouvements de lutin

```

5 REM *Programme 1*
10 CLEAR 45055: REM AFFX HEX
20 LOAD "MOVESPRITE"CODE
30 LET KEY=45056
40 LET BITPOS=45059
50 LET SPRTAB=45060
60 LET INIT=45312: REM B100 HEX
70 LET MOVSPR=45317: REM B105 HEX
80 GO SUB 1000: REM SET UP SPRITE
90 FOR I=1 TO 20
100 PRINT AT 21=RD,31=RD1:"*":
110 NEXT I
120 PRINT AT 10,16:
130 RANDOMIZE USR INIT
140 LET X=INKEY$: IF X$="" THEN GO TO 140
150 LET X=VAL X$
160 POKE KEY,X
170 RANDOMIZE USR MOVSPR
180 GO TO 140
1000 POKE BITPOS,0
1010 FOR I=0 TO 7
1020 READ X
1030 POKE SPRTAB+2+I,X
1040 POKE SPRTAB+1+2+I,0
1050 NEXT I
1060 RETURN
2000 DATA BIN 00011000
2010 DATA BIN 00011000
2020 DATA BIN 00011000
2030 DATA BIN 11111111
2040 DATA BIN 11111111
2050 DATA BIN 00011000
2060 DATA BIN 00011000
2070 DATA BIN 00011000

5 REM *Programme 2*
10 LET A=45312
20 FOR L=1000 TO 1420 STEP 10
30 S=0
40 FOR A=A TO A+7
50 READ B
60 POKE A,B
70 LET S=S+B
80 NEXT A
90 READ C
100 IF C<>S THEN PRINT "ERREUR DANS LIGNE"!!: STOP
110 NEXT L
120 READ B
130 POKE A,B
140 READ B
150 POKE (A+1),B
200 PRINT "INSEREZ BANDE PROGRAMME"
250 SAVE "MOVESPRITE"CODE 45312,400
1000 DATA 42,132,92,24,44,22,0,205,561
1010 DATA 61,177,42,1,176,58,0,176,691
1020 DATA 254,5,32,5,205,165,177,24,867
1030 DATA 24,254,6,32,5,205,109,177,812
1040 DATA 24,15,254,7,32,5,205,136,678
1050 DATA 177,24,6,254,0,192,205,228,1094
1060 DATA 177,34,1,176,22,1,205,61,677
1070 DATA 177,205,35,178,201,42,1,176,1015
1080 DATA 221,33,4,176,14,8,229,6,691
1090 DATA 2,203,66,32,6,221,126,16,672
1100 DATA 119,24,4,126,221,119,16,35,664
1110 DATA 205,83,178,221,35,16,234,225,1197
1120 DATA 13,200,221,35,197,213,205,109,1193
1130 DATA 177,209,193,24,217,62,7,164,1053
1140 DATA 254,7,48,2,36,201,17,32,589
1150 DATA 0,25,62,224,165,32,4,205,717
1160 DATA 83,178,201,124,214,7,103,201,1111
1170 DATA 62,7,164,40,2,37,201,17,530
1180 DATA 32,0,167,237,82,62,224,165,969
1190 DATA 254,224,32,4,205,76,178,201,1174
1200 DATA 124,198,7,103,201,221,33,3,890
1210 DATA 176,221,126,0,61,230,7,221,1042
1220 DATA 119,0,79,221,35,6,8,221,689
1230 DATA 94,0,221,86,1,203,3,245,853
1240 DATA 203,11,241,203,18,203,19,62,960
1250 DATA 7,185,32,3,122,83,95,221,748
1260 DATA 115,0,221,114,1,221,35,221,928
1270 DATA 35,16,220,62,7,185,192,43,760
1280 DATA 205,76,178,201,221,33,3,176,1093
1290 DATA 221,126,0,60,230,7,221,119,964
1300 DATA 0,79,221,35,6,8,221,94,664
1310 DATA 0,221,86,1,203,11,245,203,970
1320 DATA 3,241,203,26,203,27,62,0,765
1330 DATA 185,32,3,122,83,95,221,115,956
1340 DATA 0,221,114,1,221,35,221,35,848
1350 DATA 16,220,62,0,185,192,35,205,915
1360 DATA 83,178,201,42,1,176,221,33,335
1370 DATA 4,176,14,8,229,6,2,221,660
1380 DATA 126,0,47,87,126,162,221,182,951
1390 DATA 0,119,35,205,76,178,221,35,869
1400 DATA 16,237,225,13,200,197,205,109,1202
1410 DATA 177,193,24,224,62,63,188,192,1123
1420 DATA 38,87,201,62,88,188,192,38,894
1430 DATA 64,201,265

```

```

KEY EQU EB000
SPRPOS EQU EB001
BITPOS EQU EB003
SPRTAB EQU EB004
DFCC EQU E5C84
ORG EB100
INIT LD HL,(DFCC)
JR SAVSCR
MOVSPR LD D,0
CALL UNDER
LD HL,(SPRPOS)
LD A,(KEY)
CD 5
JR NZ,L0
CALL LMOVE
JR SAVSCR
L0 CP 6
JR NZ,L1
CALL BELOW
JR SAVSCR
L1 CP 7
JR NZ,L2
CALL ABOVE
JR SAVSCR
L2 CP 8
RET NZ
CALL RMOVE
SAVSCR LD (SPRPOS),HL
LD D,1
CALL UNDER
CALL PRSPRT
RET
UNDER LD HL,(SPRPOS)
LD IX,SPRTAB
LD C,8
LINE LD B,2
BYTE BIT 0,D
WIPOUT JR NZ,SVESCR
LD A,(IX+E10)
LD (HL),A
JR CONT
SVESCR LD A,(HL)
LD (IX+E10),A
CONT INC HL
INC IX
DJNZ BYTE
DEC C
RET Z
INC IX
DEC HL
DEC HL
DEC HL
PUSH BC
PUSH DE
CALL BELOW
POP DE
POP BC
JR LINE
BELOW LD A,7
AND H
CP 7
JR Z,BDIFCB
BSCAMB INC H
RET
BDIFCB LD DE,E20
ADD HL,DE
LD A,E0
AND L
RET Z
BSAMSB LD A,H
SUB 7
LD H,A
RET
ABOVE LD A,7
AND H
JR Z,ADIFCB
ASAMCB DEC H
RET
ADIFCB LD DE,E20
AND A
SBC HL,DE
LD A,E0
AND L
CP E0
RET Z
ASAMSB LD A,H
ADD A,7
LD H,A
RET
LMOVE LD IX,BITPOS
LD A,(IX+0)
DEC A
AND 7
LD (IX+0),A
LD C,A
RMOVE LD IX,BITPOS
LD A,(IX+0)
INC A
AND 7
LD (IX+0),A
LD C,A
RPAIR LD E,(IX+0)
LD D,(IX+1)
RRC E
PUSH AF
RLC E
POP AF
RR D
RR E
LD A,0
CP C
JR NZ,RSTORE
LD A,D
LD D,E
LD E,A
RSTORE LD (IX+0),E
LD (IX+1),D
INC IX
INC IX
DJNZ RPAIR
LD A,0
CP C
RET NZ
PRSPRT LD HL,(SPRPOS)
LD IX,SPRTAB
LD C,8
PRLINE LD B,2
PRBYTE LD A,(IX+0)
CPL
LD D,A
LD A,(HL)
AND D
OR (IX+0)
LD (HL),A
INC HL
INC IX
DJNZ PRBYTE
DEC C
RET Z
DEC HL
DEC HL
PUSH BC
CALL BELOW
POP BC
JR PRLINE
LD B,8
LPAIR LD E,(IX+0)
LD D,(IX+1)
RLC E
PUSH AF
RRC E
POP AF
POP RD
RL D
RL E
LD A,7
CP C
JR NZ,LSTORE
LD A,D
LD D,E
LD E,A
LSTORE LD (IX+0),E
LD (IX+1),D
INC IX
INC IX
DJNZ LPAIR
LD A,7
CP C
RET NZ
DEC HL
RET

```

## Comment utiliser ces programmes

1. Taper et sauvegarder le prog. 1, l. 10.
2. Taper et faire tourner le prog. 2 : il sauvegarde le langage machine de la mémoire, après le prog. 1.
3. Charger le prog. 1 qui se déclenche.



# Contact radio

**Motorola Inc. est l'un des plus gros producteurs de composants électroniques. Ses usines européennes et américaines fabriquent des microprocesseurs destinés aux ordinateurs 16 bits.**



Robert Galvin,  
président de Motorola.

Comme bien des entreprises florissantes, Motorola fut créée par un seul homme. Paul Galvin fonda en 1928 la Galvin Manufacturing Corporation à Chicago, et se lança dans la fabrication de postes de radio. Durant les années trente, il diversifia ses activités, passant aux appareils destinés à la police ou aux automobilistes. Au cours de la décennie suivante, la compagnie — devenue Motorola Incorporated — fut l'une des premières à produire des semi-conducteurs.

En 1959 Paul Galvin mourut, et son fils Robert lui succéda à la tête de la société. Au cours des années suivantes, d'autres constructeurs, souvent japonais, firent à Motorola une concurrence de plus en plus acharnée. La crise économique mondiale du milieu des années soixante-dix entraîna pour la firme d'énormes pertes et la contraignit à un changement complet de stratégie. On embaucha de nouveaux responsables (dont beaucoup étaient d'anciens de Texas Instruments, le vieux rival), et il fut décidé d'abandonner l'électronique traditionnelle, où il était désormais impossible de lutter, pour se concentrer sur les technologies avancées concernant la micro-électronique.

Divers secteurs d'activité furent alors vendus (celui des télévisions couleurs, par exemple) et Motorola investit massivement dans la recherche. Dans le même temps, l'entreprise rachetait d'autres sociétés installées dans des domaines où elle comptait s'implanter.

Le pari semble avoir été payant. La firme resta d'abord loin derrière les grands du marché

des semi-conducteurs, mais des investissements massifs lui ont permis de talonner le leader incontesté en ce domaine : Texas Instruments. Comme le dit Robert Galvin : « Nos concurrents ont disparu parce qu'ils n'ont pas su s'adapter. »

Motorola a pourtant eu bien des problèmes. Peu après 1975, alors que la micro-informatique était encore dans l'enfance, son microprocesseur 8 bits, le 6800, fut nettement surclassé par le 6502 de Mostek (celui du mythique Apple II), ainsi que par le 8085 d'Intel et le Z80 de Zilog, utilisés par les ordinateurs tournant sous CP/M. La firme mit sur le marché en 1976 le 6809; mais bien qu'on y reconnaisse généralement le meilleur 8 bits existant, la course pour le marché de masse était déjà perdue; seules quelques machines comme le Colour Computer de Tandy et le Dragon l'adoptèrent.

La poursuite d'une politique de gros investissements de recherche — « pour obtenir au plus vite un avantage maximum », dit Robert Galvin — a permis toutefois à Motorola d'être en bien meilleure position dans la lutte pour la domination du marché des 16 bits. Lancé en 1979 (mais réellement disponible en 1982 seulement), le 68000 a été adopté par Apple pour ses ordinateurs Lisa et Macintosh. C'est un microprocesseur très puissant, disposant de 17 registres de 32 bits, d'un bus de données de 16 bits et d'un bus d'adressage de 24 bits.

Motorola poursuit la mise au point de nouveaux produits dans ses centres de recherche de Phoenix (Arizona), de Genève, et d'East Kilbride, en Écosse. C'est dans ce dernier lieu que sont fabriquées les puces MOS (Metal Oxide Semiconductors) et CMOS (« MOS complémentaires ») destinées à de nombreux usages. La compagnie est actuellement divisée en cinq groupes : communication, semi-conducteurs, systèmes d'information, électronique automobile et industrielle, projets gouvernementaux. La rentabilité de certains secteurs reste parfois faible, mais cette firme paraît aujourd'hui en mesure de conserver la solide position qu'elle s'est acquise.

L'évolution très rapide que connaît le marché des microprocesseurs ne permet pas d'augurer de l'avenir de Motorola. Néanmoins, l'accroissement des investissements destinés à la recherche reste le meilleur atout entre les mains des responsables de la firme. Les résultats financiers de l'entreprise dépendront, quant à eux, de l'intensité de la concurrence.

Le siège social de  
Motorola, dans l'Illinois,  
USA.



# L'ENCYCLOPÉDIE DES ARMES

LES FORCES ARMÉES DU MONDE

## Pour comprendre ces terribles Armes qui font trembler le monde.

- **Des documents historiques et d'actualité :**

Vous saurez tout sur les combats de l'Intrepid américain... sur les Phantom au Vietnam... sur la guerre des Malouines du Tchad ou du Liban...

- **Des photos, des plans techniques détaillés sur les armes militaires les plus secrètes.**

Les technologies les plus avancées sur les armes offensives ou dissuasives, leur mode d'emploi, leurs performances, leur terrain d'application vous sont expliqués en détail et en maquettes.

- **Des analyses d'experts sur les forces en présence dans le monde...**

Vous connaîtrez la puissance d'armement de la Chine, de l'URSS, ou des USA... celle de Cuba, comme celle d'Israël, de la Lybie ou de l'Iran.

- **Des croquis détaillés de chaque arme passée au crible sur toutes les faces.**

Semaine après semaine, vous

pourrez constituer la plus extraordinaire panoplie de documents et d'écorchés des armes de guerre du 20<sup>e</sup> siècle.



Avec plus de 6000 photos et dessins en couleurs et 200 planches techniques, semaine après semaine, l'ENCYCLOPÉDIE DES ARMES constituera une collection aussi indispensable à l'historien, au modéliste, au journaliste qu'à tous ceux qui veulent comprendre l'enjeu des combats d'hier et d'aujourd'hui.

EDITIONS  
ATLAS



EDITIONS  
**ATLAS**